

Razvoj prototipne aplikacije za učenje engleskog jezika uporabom tehnologije proširene stvarnosti

Šanjek, David

Undergraduate thesis / Završni rad

2023

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:110:929467>

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

Download date / Datum preuzimanja: **2024-12-19**



Repository / Repozitorij:

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



MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

David Šanjek

**RAZVOJ PROTOTIPNE APLIKACIJE ZA UČENJE
ENGLSKOG JEZIKA UPORABOM TEHNOLOGIJE
PROŠIRENE STVARNOSTI**

ZAVRŠNI RAD

Čakovec, 2023.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

David Šanjek

**DEVELOPMENT OF A PROTOTYPE APPLICATION
FOR LEARNING ENGLISH LANGUAGE USING
AUGMENTED REALITY TECHNOLOGY**

ZAVRŠNI RAD

Mentor:

Nenad Breslauer, v. pred.

Čakovec, 2023.

Čakovec, 16. veljače 2023.

Polje: **2.09 Računarstvo**

ZAVRŠNI ZADATAK br. 2022-RAČ-R-8

Pristupnik: **David Šanjek (0313025583)**
Studij: Redoviti preddiplomski stručni studij Računarstvo
Smjer: Inženjerstvo računalnih sustava i mreža

Zadatak: **Razvoj prototipne aplikacije za učenje engleskog jezika uporabom tehnologije proširene stvarnosti**

Opis zadatka:

Zadatak je napraviti prototipnu aplikaciju za učenje engleskog jezika uporabom tehnologije proširene stvarnosti. Izraditi scenu te sve potrebne elemente, koristiti mogućnosti koje pruža podsustav za osvjetljenje scene, implementaciju programskog rješenja realizirati u Unity razvojnoj okolini uz pomoć Vuforia biblioteka. Aplikacija je namijenjena olakšavanju učenja stranih jezika te omogućava korisnicima interakciju s virtualnim objektima u stvarnom vremenu. Koristiti platformu Unity, programski jezik C# te dodatne programske alate. Završni rad mora sadržavati sažetak, sadržaj i uvod, nakon čega slijedi poglavlje u kojem je potrebno navesti i pojasniti osnovne ciljeve rada te očekivani rezultat. U narednom poglavlju potrebno je opisati primijenjene postupke, alate i metode. Poglavlje koje slijedi obrađivat će postignute rezultate nakon čega slijedi poglavlje u kojem se kritički raspravlja o primijenjenim metodama i postupcima te se u narednom poglavlju iznose glavni zaključci rada. Rad se završava poglavljima s popisom literature te prilogima

Rok za predaju rada: 20. rujna 2023.

Mentor:

Predsjednik povjerenstva za
završni ispit:

Breslauer

Nenad Breslauer, v. pred.

SAŽETAK:

Ovaj završni rad istražuje izradu prototipne aplikacije za učenje engleskog jezika korištenjem tehnologije proširene stvarnosti (AR) u razvojnoj okolini Unity uz pomoć biblioteke Vuforia. Cilj je rada olakšati proces učenja engleskog jezika interakcijom s virtualnim objektima u stvarnom vremenu. Aplikacija je osmišljena kako bi potaknula aktivan angažman korisnika omogućujući im istraživanje i praktično učenje engleskog jezika simuliranim scenarijima i zadacima. Aplikacija omogućuje prikazivanje objekata na temelju slike (markera), prikaz teksta koji opisuje objekt na engleskom jeziku te izgovor riječi na engleskom jeziku. Nadalje, aplikacija sadrži test kojim će se provjeriti znanje korisnika. U uvodnom dijelu rada naglašava se važnost znanja stranih jezika u globalnom društvu, a posebice engleskog jezika. Tehnologija proširene stvarnosti predstavljena je kao inovativni alat koji može obogatiti proces obrazovanja i pomoći učenicima da lakše usvajaju novi sadržaj. U prvom poglavlju postavljaju se osnovni ciljevi završnog rada i izgradnje aplikacije. Također, navodi se očekivani rezultat – funkcionalni prototip koji omogućuje korisnicima stvaranje pozitivnog i korisnog iskustva učenja engleskog jezika. U narednom poglavlju detaljno su opisani postupci, alati i metode koji su korišteni za izradu aplikacije. Platforma Unity i biblioteke Vuforia odabrane su za stvaranje AR iskustva na mobilnim uređajima, uz naglasak na interakciju s virtualnim objektima i simulirane scenarije za učenje jezika. U poglavlju posvećenom postignutim rezultatima demonstrira se funkcioniranje prototipa i prikazuju se snimke zaslona ili videozapisi koji ilustriraju aplikaciju u stvarnom vremenu. Kritička analiza primijenjenih metoda i postupaka slijedi u sljedećem poglavlju gdje se ističu pozitivni aspekti pristupa i razmatraju moguće alternativne strategije za budući razvoj aplikacije. Ističu se prednosti aplikacije, ali se također raspravlja o izazovima i mogućim poboljšanjima. U zaključnom poglavlju sumiraju se glavni zaključci završnoga rada i ističe se doprinos ovog rada području obrazovanja i primjene tehnologije proširene stvarnosti. Također, naglašava se potreba za daljnjim istraživanjem i razvojem AR aplikacija u edukacijskom kontekstu. Rad se zaključuje popisom literature koja je korištena za izradu rada.

Ključne riječi: proširena stvarnost (AR), razvojna okolina Unity, biblioteke Vuforia, engleski jezik

Sadržaj

1. Uvod	5
2. Ciljevi rada i očekivani rezultat	6
3. Opis primijenjenih postupaka, alata i metoda	7
3.1. Korištene tehnologije.....	7
3.1.1. Platforma Unity.....	7
3.1.2. Biblioteke Vuforia.....	7
3.1.3. Programski jezik C#.....	7
3.1.4. Visual Studio 2019	7
3.2. Izrada projekta	8
3.2.1. Stvaranje licencije i ključa	8
3.2.2. Izrada markera.....	9
3.2.3. Postavljanje objekata na markere (<i>Image Targete</i>).....	13
3.2.4. Dodavanje zvuka.....	16
3.2.5. Izrada početnog zaslona	19
3.2.6. Izrada provjere znanja	25
3.2.7. Izgled stranice provjere znanja	25
3.2.8. <i>QuizData</i> i <i>Scriptable Object</i>	26
3.2.9. Sustav bodovanja	32
3.2.10. Završna obrada.....	34
4. Postignuti rezultati.....	38
5. Kritički osvrt	42
6. Zaključak.....	43
Popis literature.....	44
Popis slika	46
Popis kodova	48

1. UVOD

U današnjem globaliziranom svijetu znanje stranih jezika postaje ključna vještina za ostvarivanje uspješne međunarodne komunikacije, obrazovanja i poslovnih prilika. Među tim jezicima engleski jezik zauzima posebno važno mjesto što ga čini neizostavnim alatom za povezivanje ljudi diljem svijeta. No učenje stranog jezika može biti izazovno i zahtjevno, posebice kada se radi o učenju vokabulara i gramatičkih struktura. [1,2,3]

U posljednjem desetljeću tehnologija proširene stvarnosti (AR) postala je sveprisutna i pružila je revolucionarne mogućnosti za interaktivnost i obrazovne svrhe. Integracija AR tehnologije u edukaciju može značajno unaprijediti proces učenja i potaknuti učenike da bolje razumiju i primijene novi sadržaj. [1,2,3,4,5,6,7] Upravo u tom kontekstu ovaj rad istražuje izradu prototipne aplikacije za učenje engleskog jezika tehnologijom proširene stvarnosti.

Cilj je ovog rada olakšati proces učenja engleskog jezika korisnicima interakcijom s virtualnim objektima u stvarnom vremenu. Aplikacija će se koristiti razvojnim okruženjem Unity i bibliotekom Vuforia za stvaranje AR iskustva na mobilnim uređajima. Ovom aplikacijom korisnici će imati priliku istraživati i interaktivno učiti engleski jezik simuliranim scenarijima i zadacima koji potiču aktivan angažman.

Poglavlje koje slijedi opisuje osnovne ciljeve ovog rada i očekivani rezultat. Nakon toga detaljno će se objasniti postupci, alati i metode korišteni za izradu prototipne aplikacije. Poglavlje o postignutim rezultatima pokazat će praktičnu primjenu aplikacije i analizirati njezinu učinkovitost. Slijedi kritička analiza primijenjenih metoda i postupaka kako bi se istaknule prednosti i izazovi tog pristupa.

Ovim radom želi se doprinijeti području obrazovanja i tehnologije proširene stvarnosti pružajući inovativno rješenje za učenje stranih jezika. U konačnici, nada je da će ovaj prototip poslužiti kao temelj za daljnji razvoj AR aplikacija u edukacijskom kontekstu te da će potaknuti daljnje istraživanje i primjenu tehnologije proširene stvarnosti u obrazovanju.

2. CILJEVI RADA I OČEKIVANI REZULTAT

Ovaj rad ima cilj izraditi prototipnu aplikaciju za učenje engleskog jezika pomoću tehnologije proširene stvarnosti (AR) u razvojnom okruženju Unity uz uporabu biblioteke Vuforia. Glavni ciljevi ovog istraživanja jesu olakšati proces učenja engleskog jezika interakcijom s virtualnim objektima u stvarnom vremenu te potaknuti korisnike na aktivno sudjelovanje u edukaciji i učenju.

Specifični ciljevi uključuju:

- razviti funkcionalni prototip aplikacije koji će omogućiti korisnicima istraživanje i interakciju s virtualnim objektima na mobilnim uređajima
- pružiti korisnicima mogućnost sudjelovanja u simuliranim scenarijima i zadacima za praktično učenje engleskog jezika
- personalizirati iskustvo učenja kako bi se prilagodilo individualnim potrebama i razinama znanja jezika
- motivirati korisnike za redovito učenje izazovima i rješavanjem testova
- istražiti mogućnosti primjene tehnologije proširene stvarnosti u obrazovnom kontekstu i ocijeniti njezinu učinkovitost u poticanju interakcije i boljeg usvajanja znanja.

Očekivani rezultat ovog rada jest funkcionalan prototip aplikacije koji će omogućiti korisnicima pozitivno i korisno iskustvo učenja engleskog jezika putem tehnologije proširene stvarnosti. Aplikacija će korisnicima pružiti mogućnost istraživanja i interakcije s virtualnim objektima te sudjelovanje u interaktivnim zadacima i igrama koje potiču praktičnu primjenu jezičnih sadržaja.

Ovim radom želi se demonstrirati potencijal tehnologije proširene stvarnosti u području obrazovanja i učenja jezika te potaknuti daljnja istraživanja i primjenu ove tehnologije kako bi se unaprijedilo obrazovanje i stvaranje interaktivnih edukativnih iskustava.

3. OPIS PRIMIJENJENIH POSTUPAKA, ALATA I METODA

3.1. Korištene tehnologije

U izradi prototipne aplikacije za učenje engleskog jezika putem tehnologije proširene stvarnosti koristi se razvojna okolina Unity uz pomoć biblioteke Vuforia te programski jezik C#. Koristi se i mobilna platforma Android, a testirano je na uređaju SAMSUNG GALAXY A12.

3.1.1. Platforma Unity

Unity je popularno integrirano razvojno okruženje (IDE) i igraći *engine* koji omogućuje razvoj aplikacija, igara i simulacija za različite platforme uključujući mobilne uređaje i računala. Unity je široko korištena platforma za razvoj aplikacija proširene stvarnosti zbog svoje prilagodljivosti i podrške za razne uređaje. [10,14,15]

3.1.2. Biblioteke Vuforia

Vuforia je biblioteka koja omogućuje dodavanje tehnologije proširene stvarnosti u Unityjeve projekte. Pruža napredne značajke za prepoznavanje oblika, praćenje ciljeva i stvaranje virtualnih objekata u stvarnom okruženju. Vuforia je vrlo popularna među programerima AR aplikacija zbog svoje pouzdanosti i jednostavne integracije. U ovom radu koristiti će se 2D slike koje Vuforia koristi kao markere temeljene na značajkama i detaljima 2D slika. Da bi slika bila dobro raspoznata, potrebno je da sadrži dovoljno detalja. Kada je slika jednom detektirana, Vuforia Engine nastavit će je pratiti sve dok je barem djelomično vidljiva u vidnom polju kamere. Osim 2D slika, Vuforia podražava i 3D objekte, te kuboidne objekte. [8,9,11]

3.1.3. Programski jezik C#

C# je programski jezik koji se koristi u razvojnoj okolini Unity za pisanje koda i implementaciju funkcionalnosti aplikacije. C# je popularan jezik za razvoj igara i aplikacija te ima široku podršku u platformi Unity. [12]

3.1.4. Visual Studio 2019

Visual Studio 2019 integrirano je razvojno okruženje (IDE) koje se koristi za razvoj i pisanje koda u programskom jeziku C# i ostalim jezicima. To je popularno razvojno okruženje za razvoj aplikacija i omogućuje programerima ugodno okruženje za pisanje koda, *debugiranje* i upravljanje projektima. Pruža snažne alate za pisanje i uređivanje koda te omogućuje integraciju s razvojnom okolinom Unity. [13]

3.2. Izrada projekta

Prvi korak koji treba poduzeti jest stvaranje novog Unityjeva 3D projekta. Za ostvarivanje toga treba otvoriti *Unity Hub*, a zatim odabrati opciju *New Project*. Projektu je potrebno dodijeliti naziv (npr. *MojProjekt*) te nakon toga pritisnuti *Create Project*. Nakon što je projekt uspješno stvoren, potrebno je integrirati platformu Vuforia kako bi virtualna stvarnost mogla biti iskorištena unutar projekta. Za postizanje tog cilja posjet stranici Vuforije neophodan je korak. Potrebno je izraditi besplatni korisnički račun, a potom pristupiti sekciji *Downloads*. Ondje se može preuzeti odgovarajući paket namijenjen Unityjevoj integraciji. Da bi se biblioteke Vuforia implementirale unutar okruženja Unity, treba slijediti sljedeće korake: u izborniku *Assets* odabrati opciju *Import Package*, a zatim odabrati ranije preuzeti projekt. Završni je korak potvrda uvoza pritiskom na *Import*. Na taj način projekt će biti pripremljen za daljnji razvoj i rad.

3.2.1. Stvaranje licencije i ključa

U okviru procesa izrade projekta prvobitni korak podrazumijeva konfiguraciju markera s ciljem njihova prepoznavanja. Na platformi Vuforia neophodno je generirati ključ te pribaviti licenciju za projekt Unity. Za početak potrebno je otvoriti Vuforijin *online* portal i pristupiti sekciji *Develop*, a zatim odabrati *License Manager*. Iz ponuđenih opcija odabere se opcija *Get basic* (jer je besplatna) te se upiše naziv projekta, primjerice „AR demo“. Nakon potvrde unosa licencija će biti uspješno stvorena.

Nakon generiranja ključa i licencije otvara se pravo besplatnog korištenja lokalne baze podataka s kapacitetom do 1000 markera. Također, pristup je omogućen opciji *Target Manager*. Ta ključna opcija omogućava generiranje baze podataka te učitavanje slika koje će imati ključnu ulogu kao markeri unutar aplikacije.

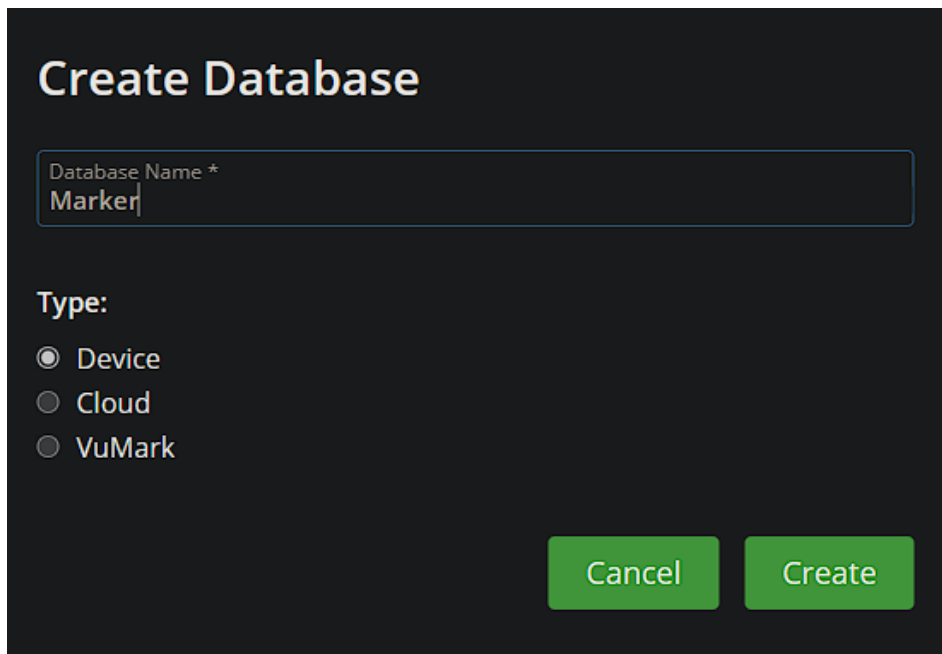
U svrhu integracije ključa i licence u okruženje Unity potrebni su sljedeći koraci:

- otvoriti projekt Unity
- u izborniku *Window* pronaći i odabrati opciju *Vuforia Configuration*
- u odjeljku *App License Key* zalijepiti prethodno kopirani ključ.

Na taj način osigurana je uspješna implementacija ključne komponente potrebne za daljnji razvoj i funkcionalnost aplikacije.

3.2.2. Izrada markera

Markeri će u ovom projektu biti u obliku 2D slika, a da bi ih se izradilo, potrebno je na Vuforiajinu *online* portalu kliknuti na *Develop* → *Target Manager* → *Add Database* i namjestiti onako kako je prikazano na slici 1.



Slika 1. Izrada baze

Izvor: autor

Nakon klika na *Create* stvorena je baza podataka. Sada joj se može pristupiti klikom na nju. *Markerima* se pristupa klikom na *Add Target* i unesu se slike po želji, na kojima će biti prikazani objekti kad ih *AR Camera* u Unityju prepozna. Jako je važna sekcija *Width* koja je izražena u metrima, a predstavlja širinu papira koja je u ovom slučaju širina A4 papira, odnosno 0,215 m. Klikom na *Add* kreirani su *Markeri*. Za potrebe ovog projekta korišteno je sedam markera: tanjur, odrezak, riba, nož, keks, kolač (torta) i jabuka.

Add Target

Type:

Image Multi Cylinder Object

File:

2e9443265ef18e9b7b002c28de8a8c9f.jpg Browse...

.jpg or .png (max file 2mb)

Width:

0.215

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

fish

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Cancel Add

Slika 2. Dodavanje Image Targeta u bazu

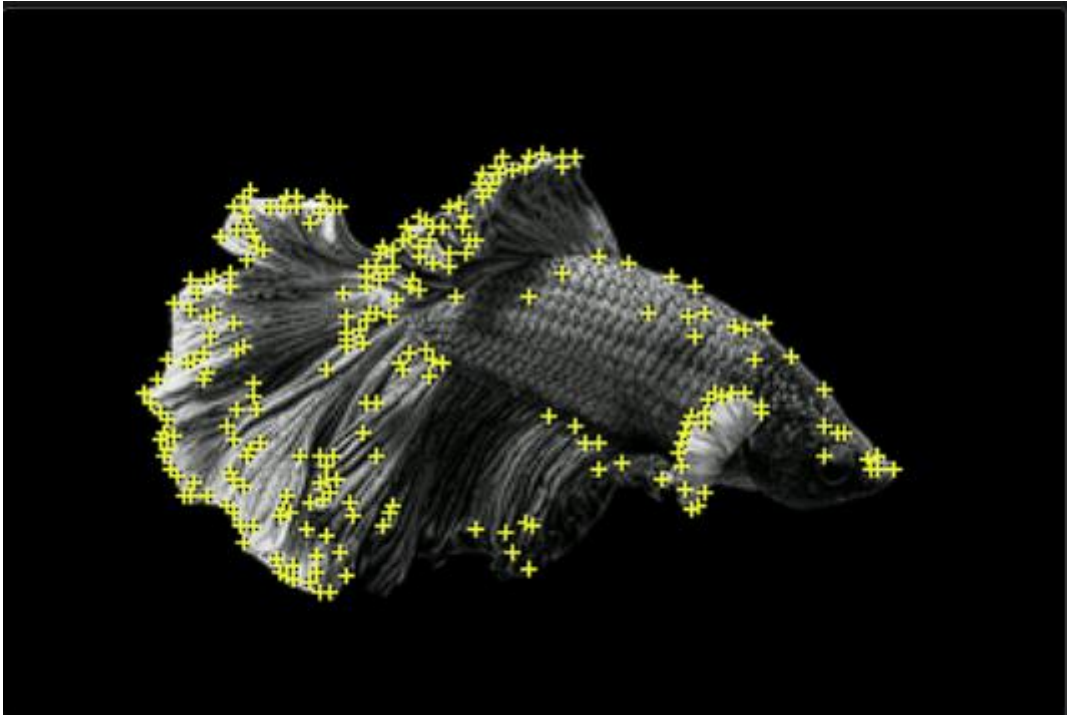
Izvor: autor

Target Name	Type	Rating [ⓘ]	Status [▼]
<input type="checkbox"/> Plate	Image	★★★★★	Active
<input type="checkbox"/> Cookie	Image	★★★★★	Active
<input type="checkbox"/> Cake	Image	★★★★★	Active
<input type="checkbox"/> Steak	Image	★★★★★	Active
<input type="checkbox"/> Apple	Image	★★★★★	Active
<input type="checkbox"/> Knife	Image	★★★★★	Active
<input type="checkbox"/> Fish	Image	★★★★★	Active

Slika 3. Prikaz svih Image Targeta (markera);

Izvor: autor

Rating predstavlja prepoznatljivost slike, odnosno promjene u teksturi ili detaljima ili neke prepoznatljive točke po kojima bi Vuforia mogla prepoznati tu sliku, primjer se može vidjeti na slici dolje. Naravno, ako je *rating* viši, Vuforia će lakše prepoznati sliku, odnosno *marker*.

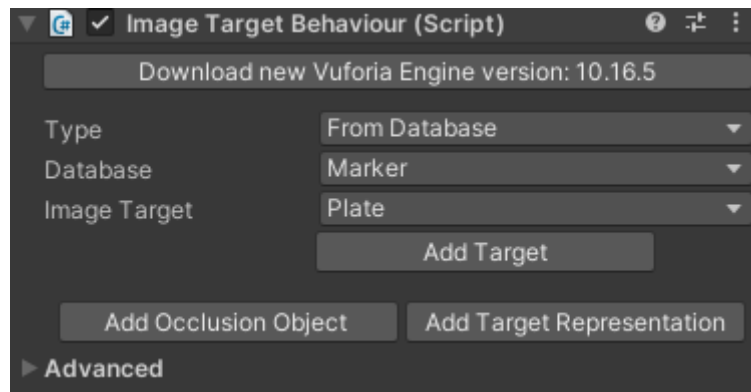


Slika 4. Prepoznatljivost slike;

Izvor: autor

U projektu Unity, kako bi se ti markeri koristili, koraci koje treba slijediti jesu sljedeći:

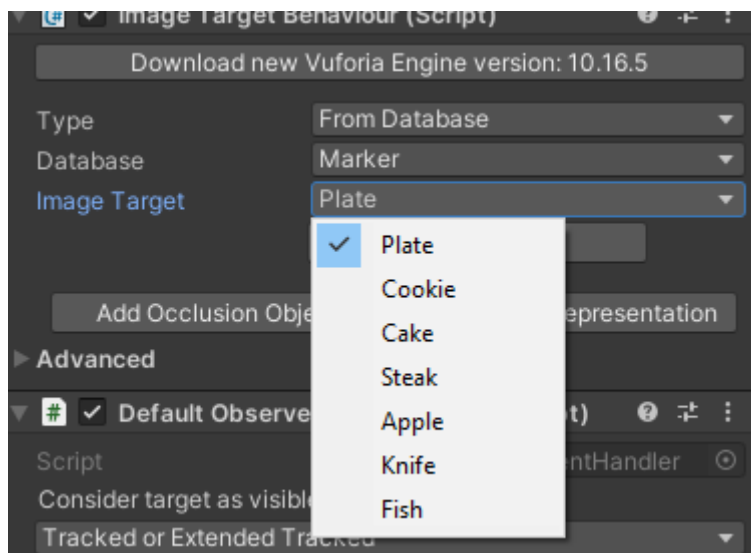
- baza podataka za markere preuzima se klikom na opciju *Download Database* unutar odjeljka s markerima te se uveze u projekt
- u hijerarhiji projekta Unity kontekstni izbornik otvara se desnim klikom. Odabire se *Vuforia Engine* i dodaju se komponente *AR Camera* i *Image*
- u prozoru *Inspector* konfiguracije se postavljaju onako kako je vidljivo na slici 5.



Slika 5. Dodavanje markera u Unityju;

Izvor: autor

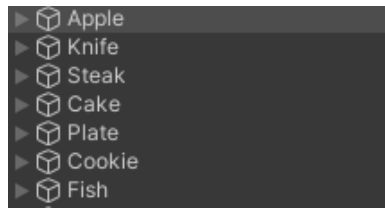
Naravno, postavljaju se svi markeri, odnosno njih sedam, samo je u odnosu na sliku potrebno promijeniti odjeljak *Image Target* (slika 6).



Slika 6. Dodavanje markera u Unityju

Izvor: autor

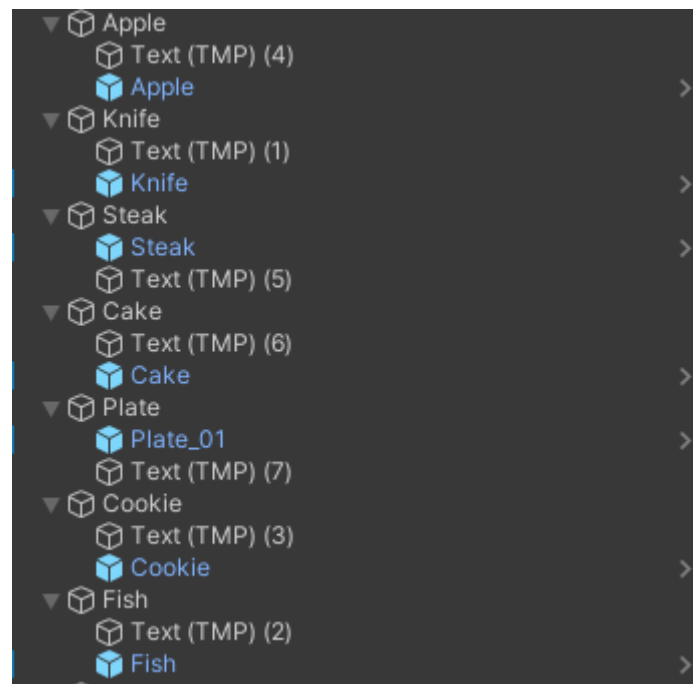
Nakon dodavanja svih sedam *Image Targeta* hijerarhija izgleda ovako:

Slika 7. Prikaz hijerarhije nakon dodavanja markera (*Targeta*)

Izvor: autor

3.2.3. Postavljanje objekata na markere (*Image Targete*)

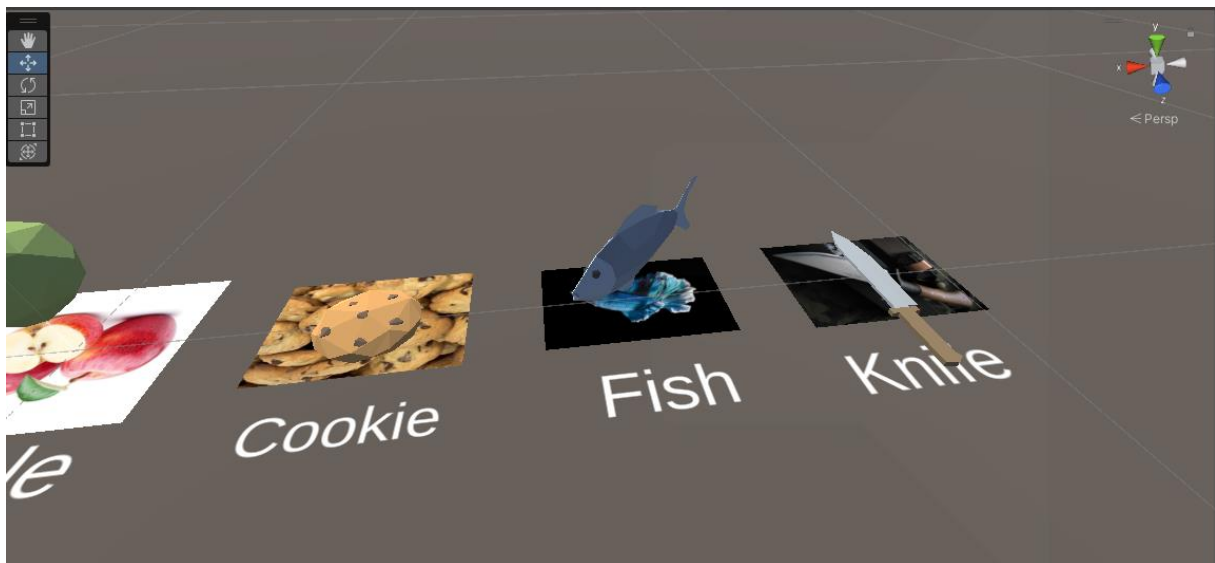
Radi postavljanja objekta na marker potrebno je odabrati željeni objekt te ga postaviti kao „dijete“ (engl. *child*) *Image Targeta*. U svrhu ovog projekta odabran je paket *Low Poly Food Lite* koji sadrži 28 objekata od kojih je sedam odabrano za daljnje korištenje. Ista je situacija i s tekstom, tekst koji se želi koristiti jednostavno se postavi kao „dijete“ željenog *Image Targeta*. Nakon izvršenih željenih radnji ovako sada izgleda hijerarhija:



Slika 8. Izgled hijerarhije nakon dodavanja teksta i objekata

Izvor: autor

A ovako izgleda scena:



Slika 9. Prikaz objekata u sceni

Izvor: autor



Slika 10. Prikaz objekata u sceni

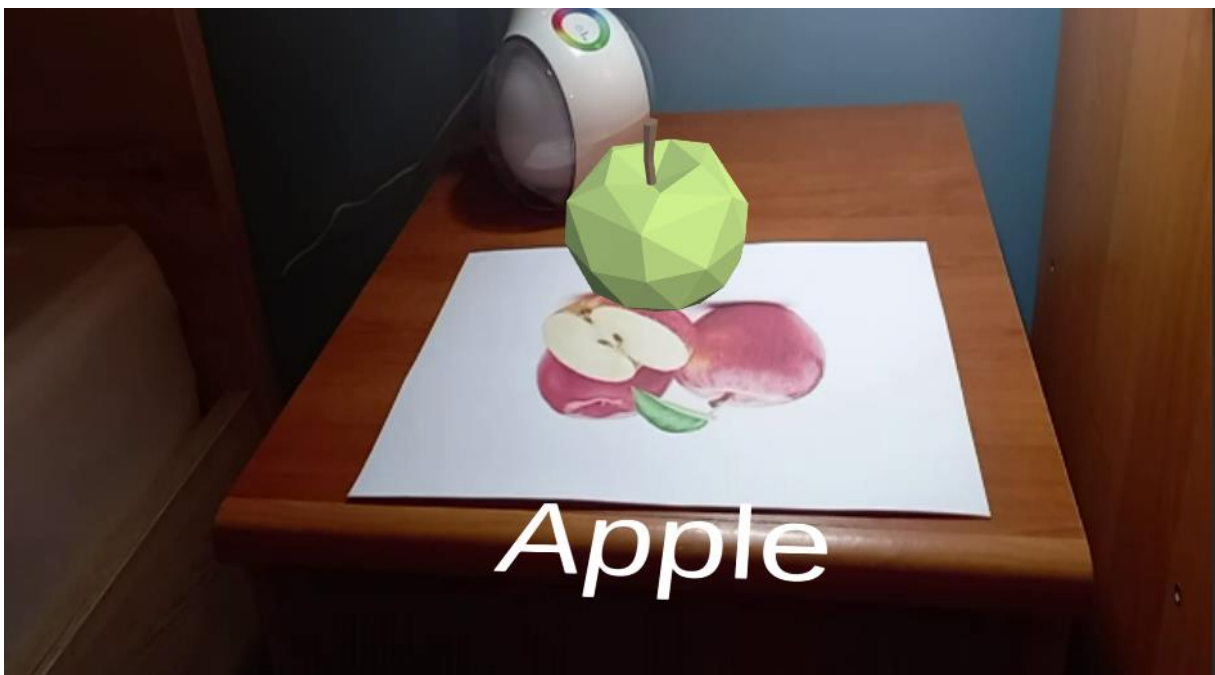
Izvor: autor

Nakon pokretanja AR kamere, odnosno projekta rezultat izgleda ovako:



Slika 11. Prikaz rada aplikacije u Unityju

Izvor: autor



Slika 12. Prikaz rada aplikacije u Unityju

Izvor: autor

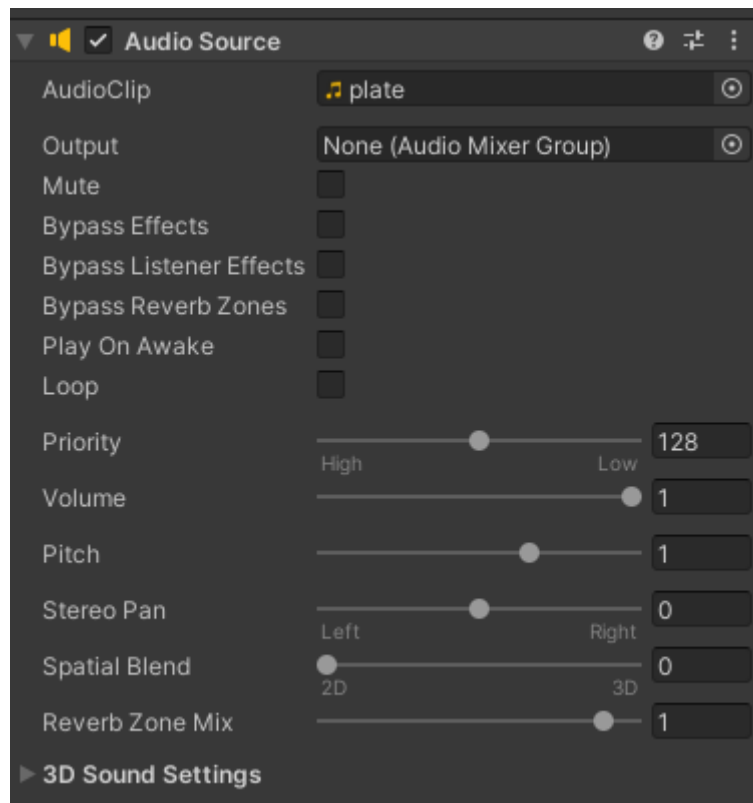


Slika 13. Prikaz rada aplikacije u Unityju

Izvor: autor

3.2.4. Dodavanje zvuka

Desnim klikom na *Image Target* otvara se kontekstualni izbornik omogućavajući pristup dodavanju zvuka. U kontekstualnom izborniku odabere se opcija *Audio* te nakon toga *Audio Source*. Time se stvara veza za dodavanje zvuka na *Image Target*. Zvuk je preuzet sa stranice *Sound of Text* koja se koristi za izgovor engleskih riječi [16]. Preuzete zvučne datoteke (.mp3) trebaju se uvesti u Unity tako da ih se jednostavno povuče u mapu *Assets*. Za svaki *Image Target* zahtijeva se dodavanje odgovarajuće zvučne datoteke. To se postiže dodavanjem zvučne datoteke kao dijela komponente *Audio Source* i postavljanjem te datoteke unutar odjeljka *Audio Clip*, odnosno zvučna datoteka za određeni *Image Target* (na slici dolje, *plate* (tanjur)) povuče se u odjeljak *Audio Clip*. Opcija *Play on Awake* mora biti isključena da se zvuk ne bi pokretao odmah prilikom pokretanja scene.



Slika 14. Dodavanje zvuka za tanjur (engl. plate)

Izvor: autor

Da bi se zvuk reproducirao kada se kamera usmjeri na sliku, potrebno je napisati skriptu (kod) koja će to konfigurirati. Za dodavanje skripte potrebno je samo desnim klikom u mapi *Assets* u izborniku kliknuti na *Create* → *C# script*. Nakon što joj se da naziv, koji će u ovom slučaju biti *PlaySoundOnTargetFound*, otvara se Visual Studio 2019 u kojem se piše kod. Kod za ovu skriptu nalazi se ispod.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;

public class PlaySoundOnTargetFound : MonoBehaviour
{
    // Referenca na izvor zvuka koji će se koristiti za reprodukciju zvuka
    public AudioSource audioSource;

    // Zvuk koji će se reproducirati kad se ciljni objekt prepozna
    public AudioClip audioClip;

    // Referenca na komponentu ObserverBehaviour koja prati status ciljnog objekta
    private ObserverBehaviour observerBehaviour;

    void Start()
    {
        // Dobivanje reference na ObserverBehaviour komponentu ovog objekta
        observerBehaviour = GetComponent<ObserverBehaviour>();

        // Provjera postoji li ObserverBehaviour komponenta i registracija događaja za praćenje promjena statusa ciljnog objekta
        if (observerBehaviour)
        {
            observerBehaviour.OnTargetStatusChanged += OnTargetStatusChanged;
        }
    }
}

```

Kod 1. Prikaz prvog dijela skripte koja predstavlja aktivaciju zvuka

Izvor : Autor

```

void OnDestroy()
{
    // Uklanjanje registracije događaja prilikom uništenja objekta
    if (observerBehaviour)
    {
        observerBehaviour.OnTargetStatusChanged -= OnTargetStatusChanged;
    }
}

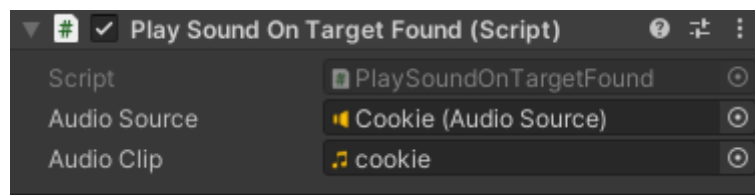
// Metoda koja se poziva kad se promijeni status ciljnog objekta
private void OnTargetStatusChanged(ObserverBehaviour behaviour, TargetStatus targetStatus)
{
    // Provjera je li status ciljnog objekta TRAČAN (TRACKED)
    if (targetStatus.Status == Status.TRACKED)
    {
        // Postavljanje željenog zvuka iz audioClip varijable i reprodukcija zvuka
        audioSource.clip = audioClip;
        audioSource.Play();
    }
}
}

```

Kod 2. Prikaz drugog dijela skripte koja predstavlja aktivaciju zvuka

Izvor : Autor

Za dodjelu skripte *Image Targetima* potrebno je označiti *Image Target* kojem se želi dodijeliti skripta. U prozoru inspektor potrebno je izabrati opciju *Add component* nakon čega se unosi ime skripte. Na taj način skripta je dodijeljena *Image Targetu*. Sada su skriptom definirana dva polja, odnosno odjeljka; u *Audio Sourceu* odabire se prije kreirani *Audio Source*, a u *Audio clipu* zvučna datoteka.



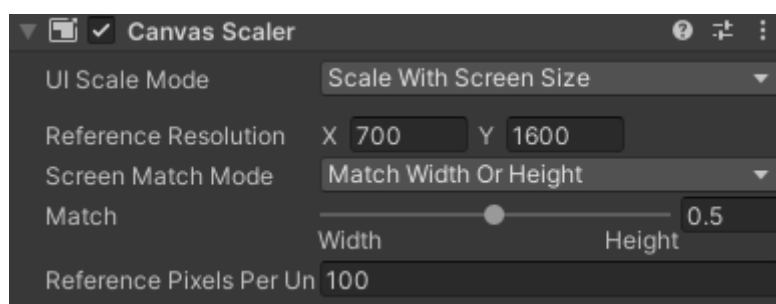
Slika 15. Prikaz dvaju polja napisanih u skripti

Izvor: autor

3.2.5. Izrada početnog zaslona

U ovom poglavlju fokus će biti na razvoju početnog ekrana koji će predstavljati prvu interakciju korisnika s ovim edukativnim alatom. Početni ekran ima ključnu ulogu u privlačenju pažnje korisnika pružajući im intuitivan i angažiran uvod u ono što aplikacija nudi. Uvodni ekran treba pružiti jasan pregled funkcionalnosti aplikacije, motivira korisnike da nastave dalje te ih inspirira da istraže mogućnosti učenja engleskog jezika različitim modulima, interaktivnim lekcijama, vježbama i testovima.

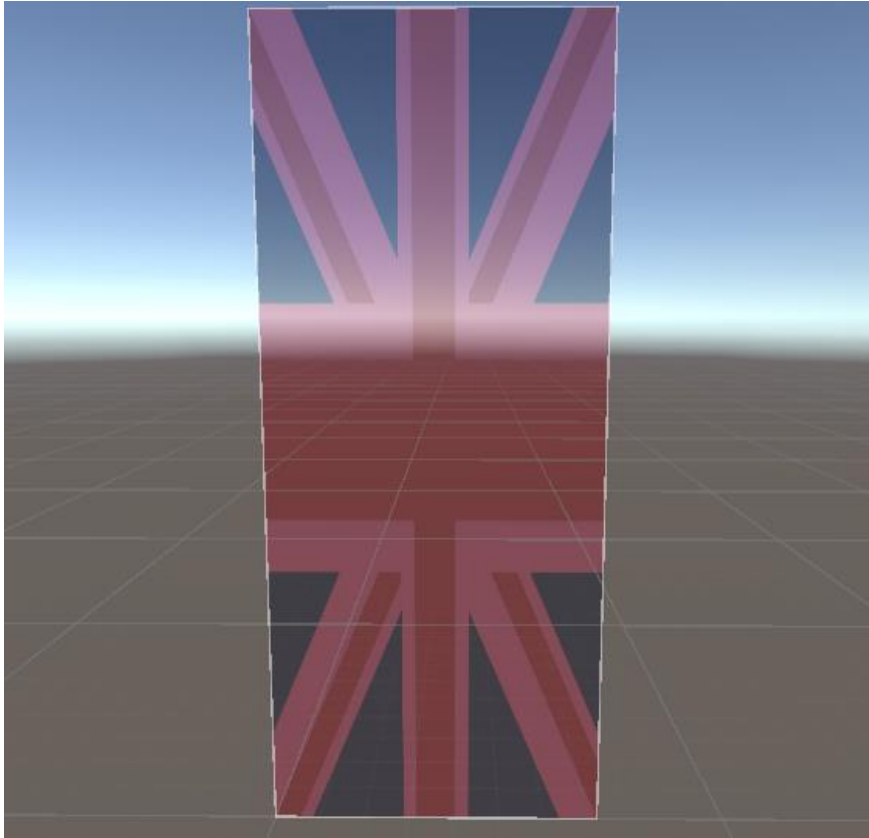
Prvo, potrebno je kreirati novu scenu. To se može napraviti na način da se klikne na *file* → *new scene*, zatim *file* → *save scene as* i doda se ime scene, u ovom slučaju *homescreen*. Dodaje se *canvas*, desnim klikom na hijerarhiju → *UI* → *Canvas*, a konfiguracija je u inspektoru sljedeća:



Slika 16. Prikaz postavki Canvasa

Izvor : Autor

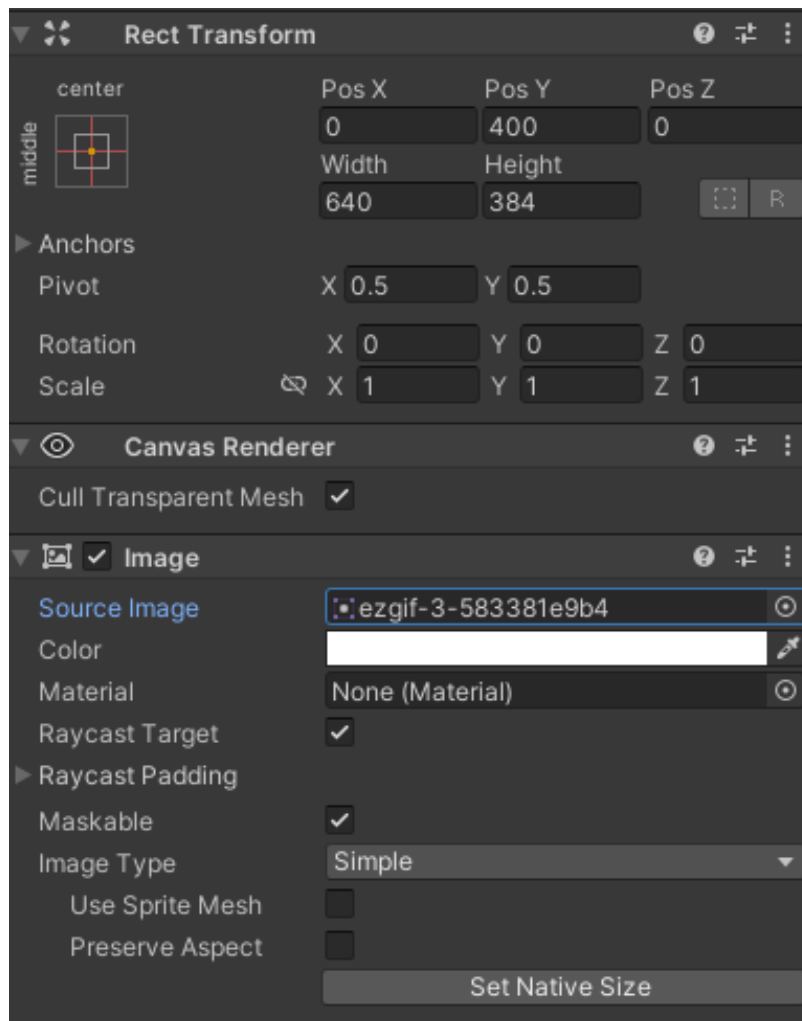
Za rezoluciju je odabrana 700 x 1600 jer uređaj na kojem je aplikacija testirana posjeduje upravo tu rezoluciju. Nakon toga dodan je panel desnim klikom na *canvas* → *UI* → *Panel* te je kao panel postavljena slika engleske zastave što se može vidjeti na slici 17.



Slika 17. Izgled panela u obliku engleske zastave

Izvor: autor

Poslije toga dodana je i slika engleske zastave na gornjem dijelu. Slika se dodaje na sljedeći način: desni klik na *canvas* → UI → *Image*. Preuzeta slika stavi se na polje *Source Image*, a sve ostalo konfigurirano je prema postavkama prikazanim na slici 18.



Slika 18. Dodavanje slike na panel

Izvor: autor

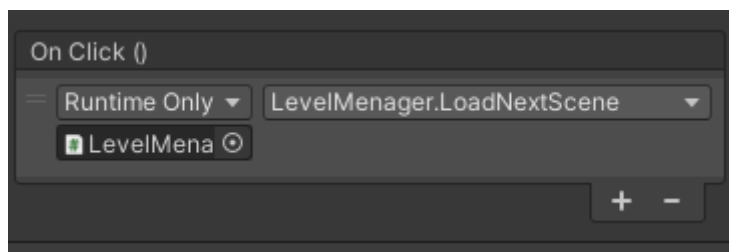
Početni zaslon sada izgleda ovako:



Slika 19. Slika na panelu

Izvor: autor

Još su dodana tri gumba koja se dodaju na sljedeći način: desni klik na *canvas* → UI → *Button*. Na svakome gumbu nalazi se tekst koji je konfiguriran po želji. Prvi gumb vodi do scene s virtualnom stvarnosti i služi za učenje, drugi gumb služi za provjeru znanja i vodi do scene s testom, a treći gumb služi za izlazak. Kako bi se to implementiralo, potrebno je napisati skriptu koja će omogućavati prijelaz scena. Naziv skripte jest *Level Menager* čiji se kod može vidjeti ispod. Kreirajući prazan *gameObject* naziva *Level Menager* stavlja se skripta na njega te se unutar inspektora na gumbе pomoću funkcije *OnClick* stavljaју željene funkcije.



Slika 20. Level Menager

Izvor: autor


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

public class LevelMenager : MonoBehaviour
{
    // Funkcija za učitavanje sljedeće scene u postavkama izgradnje
    public void LoadNextScene()
    {
        int currentSceneIndex = SceneManager.GetActiveScene().buildIndex;
        SceneManager.LoadScene(currentSceneIndex + 1);
    }

    // Funkcija za učitavanje određene scene prema njenom indeksu u postavkama izgradnje
    public void LoadSceneByIndex(int sceneIndex)
    {
        if (sceneIndex >= 0 && sceneIndex < SceneManager.sceneCountInBuildSettings)
        {
            SceneManager.LoadScene(sceneIndex);
        }
        else
        {
            Debug.LogWarning("Neispravan indeks scene!");
        }
    }

    // Funkcija za učitavanje određene scene prema njenom imenu
    public void LoadSceneByName(string sceneName)
    {
        SceneManager.LoadScene(sceneName);
    }

    // Funkcija za izlazak iz aplikacije
    public void QuitApplication()
    {
#if UNITY_EDITOR
        UnityEditor.EditorApplication.isPlaying = false;
#else
        Application.Quit();
#endif
    }
}
```

Kod 3. Level Menager

Izvor : Autor



Slika 21. Završni izgled početnog zaslona

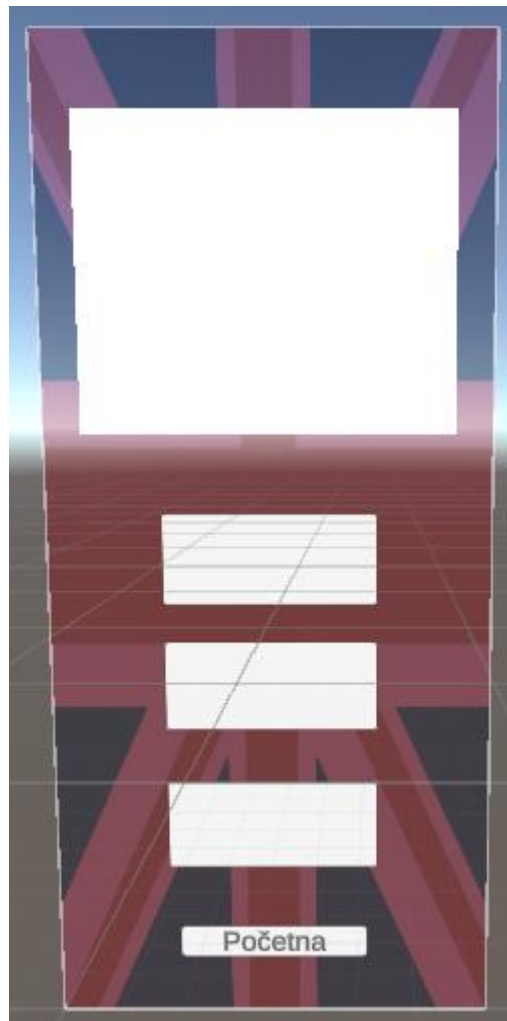
Izvor: autor

3.2.6. Izrada provjere znanja

Provjera znanja dizajnirana je kao test koji se sastoji od sedam slika koje su jednake onima pomoću kojih se prethodno moglo učiti, a ispod nje nalaze se tri gumba koji sadrže svaki po jedan odgovor. Na korisniku je da izabere točnu riječ na engleskom jeziku koja je prikazana slikom. Korisnik mora odgovoriti točno na pitanje, a ako odgovori pogrešno, provjera se svejedno nastavlja dalje do samoga kraja na kojem je prikaz bodova. Prvo je napisana skripta koja služi kao baza, sadrži slike, sve odgovore, i one točne i one netočne, pri čemu su točni definirani posebnim odjeljkom. Nakon toga je napisana skripta koja će iskoristiti te informacije iz baze i raspodijeliti ih na gumbе.

3.2.7. Izgled stranice provjere znanja

Stranica izgleda manje-više isto i dizajnirana je na isti način kao i početna, s time da tekst na gumbima nije definiran u sceni, nego će se pojavljivati tijekom izvođenja aplikacije zato što se koristi podacima definiranim u bazi, odnosno već prije spomenutoj skripti koja služi kao baza. Također se na vrhu pojavljuju slike koje će isto tako biti automatski definirane tijekom pokretanja scene koristeći se podacima iz baze.



Slika 22. Izgled zaslona koji predstavlja provjeru znanja

Izvor: autor

Gumb „Početna“ vodi na početni zaslon.

3.2.8. *QuizData* i *Scriptable Object*

Scriptable Object vrsta je skripte u Unityju koja je posebno dizajnirana za držanje različitih vrsta podataka. [14] Za razliku od tradicionalnih C# skripti, *Scriptable Object* ne moraju biti, i zapravo ne mogu biti, direktno vezani za *GameObjects*. [14] *Scriptable Object* spremnik je podataka koji se može koristiti za spremanje velikih količina podataka, neovisno o instancama klase. [15] Jedna je od glavnih upotreba *Scriptable Objecta* smanjenje upotrebe memorije projekta izbjegavanjem kopija vrijednosti. [15]

Quiz Data naziv je skripte koja će omogućiti korištenje *Scriptable Objecta* i samim time skladištenje pitanja i odgovora. Skripta izgleda ovako:

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

// Ovaj atribut omogućuje stvaranje novog objekta QuizData izravno iz Unity Editor-a
[CreateAssetMenu(fileName = "NewQuizData", menuName = "Quiz/QuizData")]

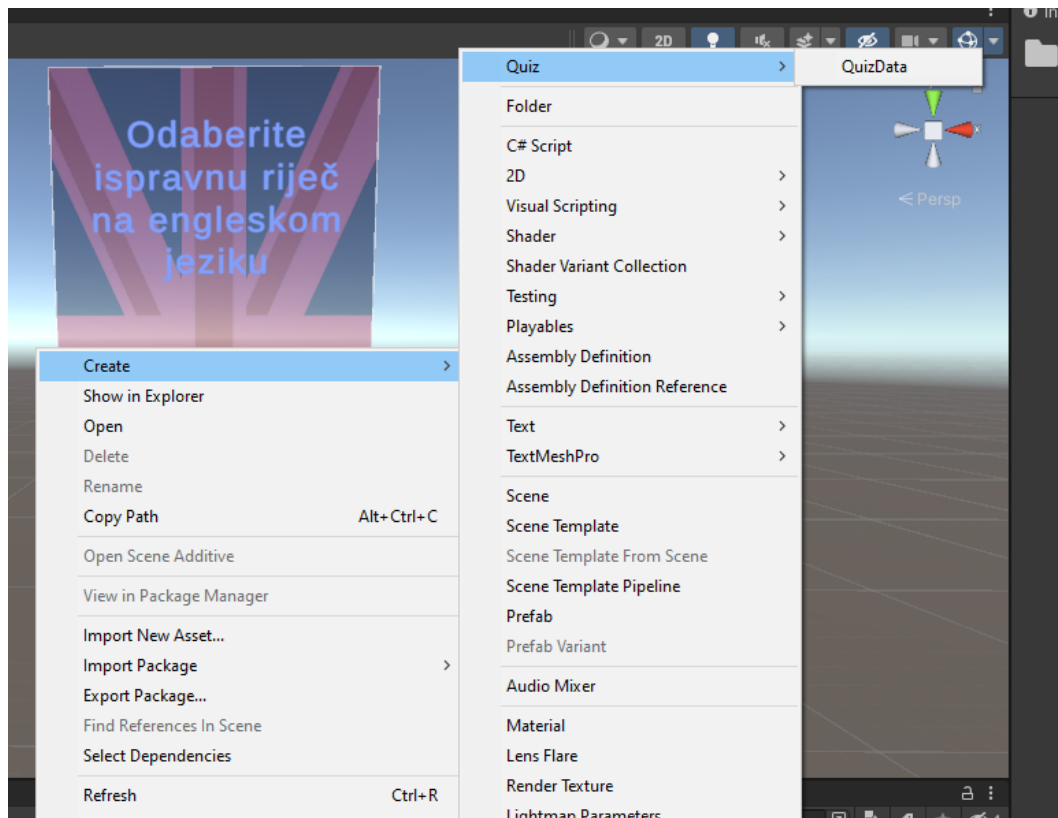
public class QuizData : ScriptableObject // Klasa QuizData nasljeđuje ScriptableObject
{
    public string[] answerOptions; // Niz opcija odgovora
    public string[] correctAnswers; // Niz točnih odgovora
    public Sprite[] images; // polje slika
}
```

Kod 4. Scriptable Object

Izvor: autor

Taj kod definira klasu *QuizData* koja nasljeđuje *Scriptable Object*. Klasa *QuizData* sadrži tri javna polja: *images*, *answerOptions* i *correctAnswers*, koji su svi nizovi *stringova*, osim *images* koji je *Sprite*. *Sprite* je 2D grafika koja se koristi u Unityju. Ta klasa može se koristiti za spremanje podataka o kvizu, uključujući pitanja, 2D grafiku (slike), opcije odgovora i točne odgovore.

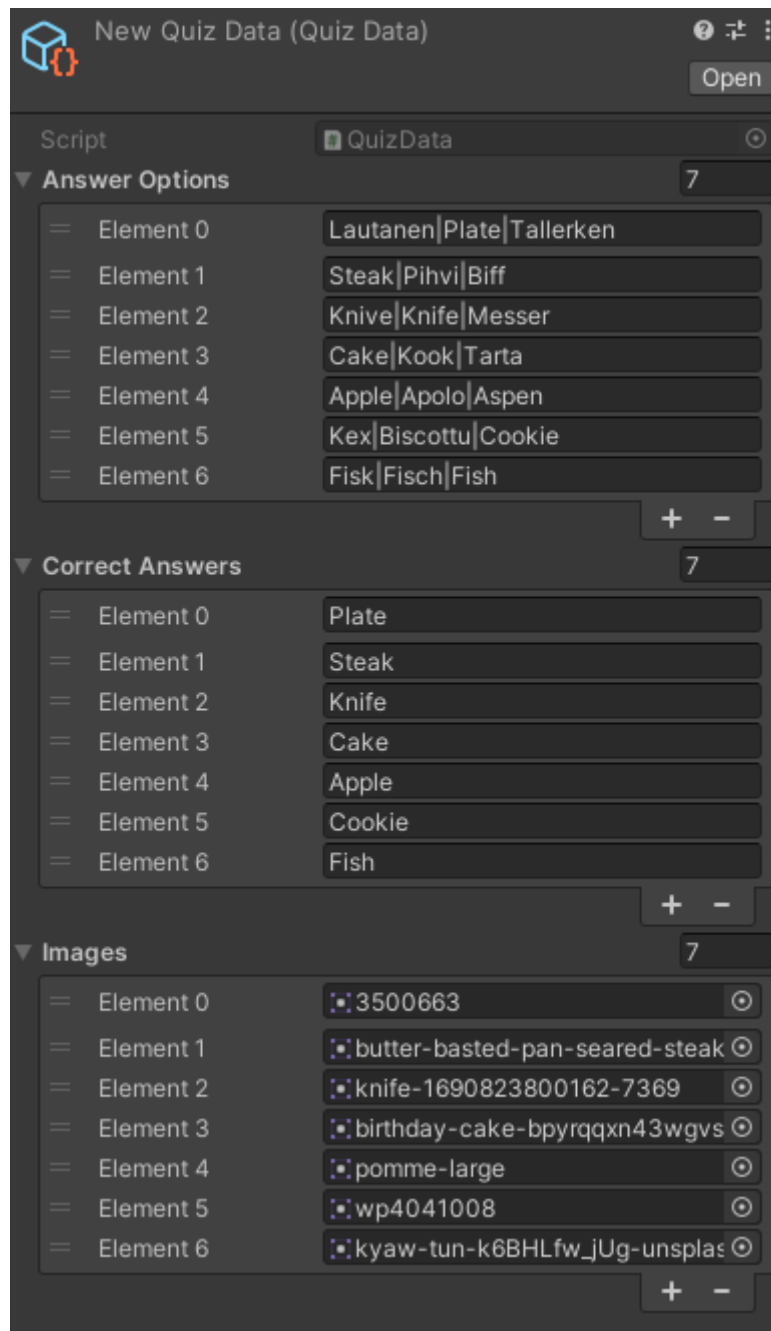
Sada se može odabrati opcija *Quiz/QuizData* u izborniku *Assets/Create* da bi se stvorio novi objekt *QuizData* s imenom datoteke *NewQuizData*.



Slika 23. Postupak stvaranja klase QuizData

Izvor: autor

Nakon klika na *QuizData* stvara se *Asset* te se u inspektoru otvara prozor kojim su definirana pitanja, opcije odgovora i točni odgovori kao na slici ispod. Slike su uvezene u Unity i postavljene pod odjeljak *Images*.



Slika 24. Dodavanje pitanja, odgovora i slika

Izvor: autor

Sada još samo ostaje napisati skriptu koja će iz *QuizData* vaditi podatke i prikazivati ih na sceni, a naziv joj je *QuizManagerScript*. Taj je kod skripta za upravljanje kvizaškom igrom u Unityju. Prikazuje slike i njihove odgovarajuće opcije odgovora, provjerava je li dani odgovor točan i pruža povratne informacije korisniku te na kraju dodjeljuje korisniku broj bodova ovisno o tome na koliko je pitanja odgovorio točno.

```

using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
using TMPro;

public class QuizManagerScript : MonoBehaviour
{
    public Image questionImage; // Slike
    public Button[] answerButtons; // Gumbi za odgovore
    public GameObject feedbackPanel; // Panel povratnih informacija
    public QuizData Quiz; // Podaci kviza

    private int currentQuestionIndex = 0; // Trenutni indeks pitanja
    private int score = 0; // Rezultat igrača

    public void OnButtonClick(Button button)
    {
        string answerText = button.GetComponentInChildren<TextMeshPro>().text; // Tekst odgovora na koji je igrač kliknuo
        CheckAnswer(answerText); // Provjera točnog odgovora
    }

    void Start()
    {
        DisplayQuestion(currentQuestionIndex); // Prikaz prvog pitanja pri pokretanju
    }
}

```

Kod 5. Prvi dio skripte za upravljanje podacima iz klase QuizData

Izvor: autor

```

public void DisplayQuestion(int questionIndex)
{
    if (questionIndex >= 0 && questionIndex < Quiz.images.Length)
    {
        questionImage.sprite = Quiz.images[questionIndex]; // Postavljanje slike

        string[] answers = Quiz.answerOptions[questionIndex].Split('|'); // Dohvaćanje odgovora
        for (int i = 0; i < answers.Length; i++)
        {
            int randomIndex = UnityEngine.Random.Range(i, answers.Length); // Generacija slučajnog indeksa za miješanje odgovora
            string temp = answers[i];
            answers[i] = answers[randomIndex]; // Zamjena trenutnog odgovora s odgovorom na slučajnom indeksu
            answers[randomIndex] = temp;
        }

        for (int i = 0; i < answerButtons.Length; i++)
        {
            // Postavljanje teksta na gumbu na odgovarajući odgovor
            answerButtons[i].GetComponentInChildren<TextMeshPro>().text = answers[i];
        }

        feedbackPanel.SetActive(false); // Sakrij panel povratnih informacija
    }
}

```

Kod 6. Drugi dio skripte za upravljanje podacima iz klase QuizData

Izvor: autor


```

public void CheckAnswer(string answerText)
{
    string correctAnswer = Quiz.correctAnswers[currentQuestionIndex]; // Dohvaćanje točnog odgovora za trenutno pitanje

    if (correctAnswer.Contains(answerText)) // Provjera je li odgovor igrača sadržan u točnom odgovoru
    {
        ShowFeedback(true); // Prikaz povratnih informacija za točan odgovor
        score++; // Povećanje rezultata igrača
    }
    else
    {
        ShowFeedback(false); // Prikaz povratnih informacija za netočan odgovor
    }

    currentQuestionIndex++; // Prijelaz na sljedeće pitanje
    if (currentQuestionIndex < Quiz.images.Length)
    {
        Invoke("DisplayNextQuestion", 2f); // Prikaz sljedećeg pitanja nakon 2 sekunde
    }
    else
    {
        PlayerPrefs.SetInt("FinalScore", score); // Spremanje konačnog rezultata u igračevu memoriju
        SceneManager.LoadScene("Score"); // Prijelaz na scenu za prikaz rezultata
    }
}

```

Kod 7. Treći dio skripte za upravljanje podacima iz klase QuizData

Izvor: autor

```

private void DisplayNextQuestion()
{
    DisplayQuestion(currentQuestionIndex); // Prikaz sljedećeg pitanja
}

private void ShowFeedback(bool isCorrect)
{
    feedbackPanel.SetActive(true); // Prikaz panela povratnih informacija
    // Dohvati tekst za povratne informacije
    TextMeshProUGUI feedbackText = feedbackPanel.GetComponentInChildren<TextMeshProUGUI>();

    if (isCorrect)
    {
        feedbackText.text = "Točno!"; // Postavljanje teksta povratnih informacija za točan odgovor
        feedbackText.color = Color.green; // Postavljanje zelene boje teksta
    }
    else
    {
        feedbackText.text = "Netočno!"; // Postavljanje tekst apovratnih informacija za netočan odgovor
        feedbackText.color = Color.red; // Postavljanje crvene boje teksta
    }
}

```

Kod 8. Četvrti dio skripte za upravljanje podacima iz klase QuizData

Izvor: autor

3.2.9. Sustav bodovanja

Za sustav bodovanja napravljena je završna scena na kojoj će biti prikazani korisnikovi bodovi. Napisana je posebna skripta koja izvlači podatke iz skripte *QuizMenager* i prikazuje spremljene korisnikove podatke.



Slika 25. Izgled scene za prikaz bodova

Izvor: autor

Dodani su gumbi za ponavljanje provjere znanja te gumb koji vodi na početni zaslon.

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

public class ScoreSceneScript : MonoBehaviour
{
    public TextMeshProUGUI finalScoreText; // dodavanje nove varijable, final score text

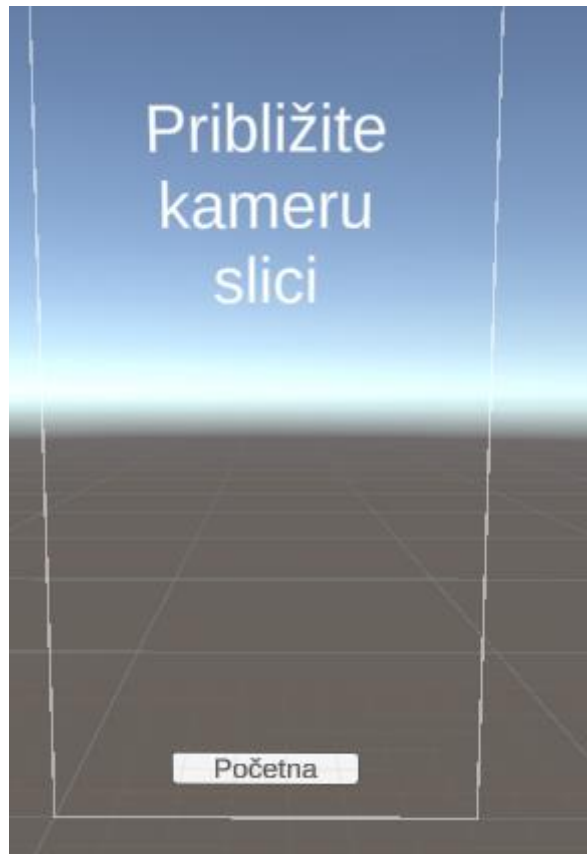
    void Start()
    {
        // Dohvaćanje konačnog rezultata pomoću PlayerPrefs iz skripte QuizMenager
        int finalScore = PlayerPrefs.GetInt("FinalScore");
        // Ažuriranje teksta konačnog rezultata
        finalScoreText.text = " " + finalScore + "/7";
    }
}
```

Kod 9. Skripta za prikaz bodova

Izvor: autor

3.2.10. Završna obrada

Za kraj još je dodan tekst u glavnoj sceni, a glasi *Približite kameru slici* i nestaje nakon tri sekunde. U sceni koja se upotrebljava za provjeru znanja dodan je još gumb *Početna* da bi se korisnik mogao vratiti na početni zaslon.



Slika 26. Tekst koji nestaje nakon tri sekunde

Izvor: autor

Da bi tekst nestao nakon nekoliko sekundi, napisana je skripta čiji je kod vidljiv ispod.

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

// Ova skripta prikazuje tekst na ekranu na određeno vrijeme
public class showText : MonoBehaviour
{
    // UI element za prikaz teksta
    public GameObject textPanel;

    // Trajanje prikaza teksta u sekundama
    public float displayDuration = 3f;

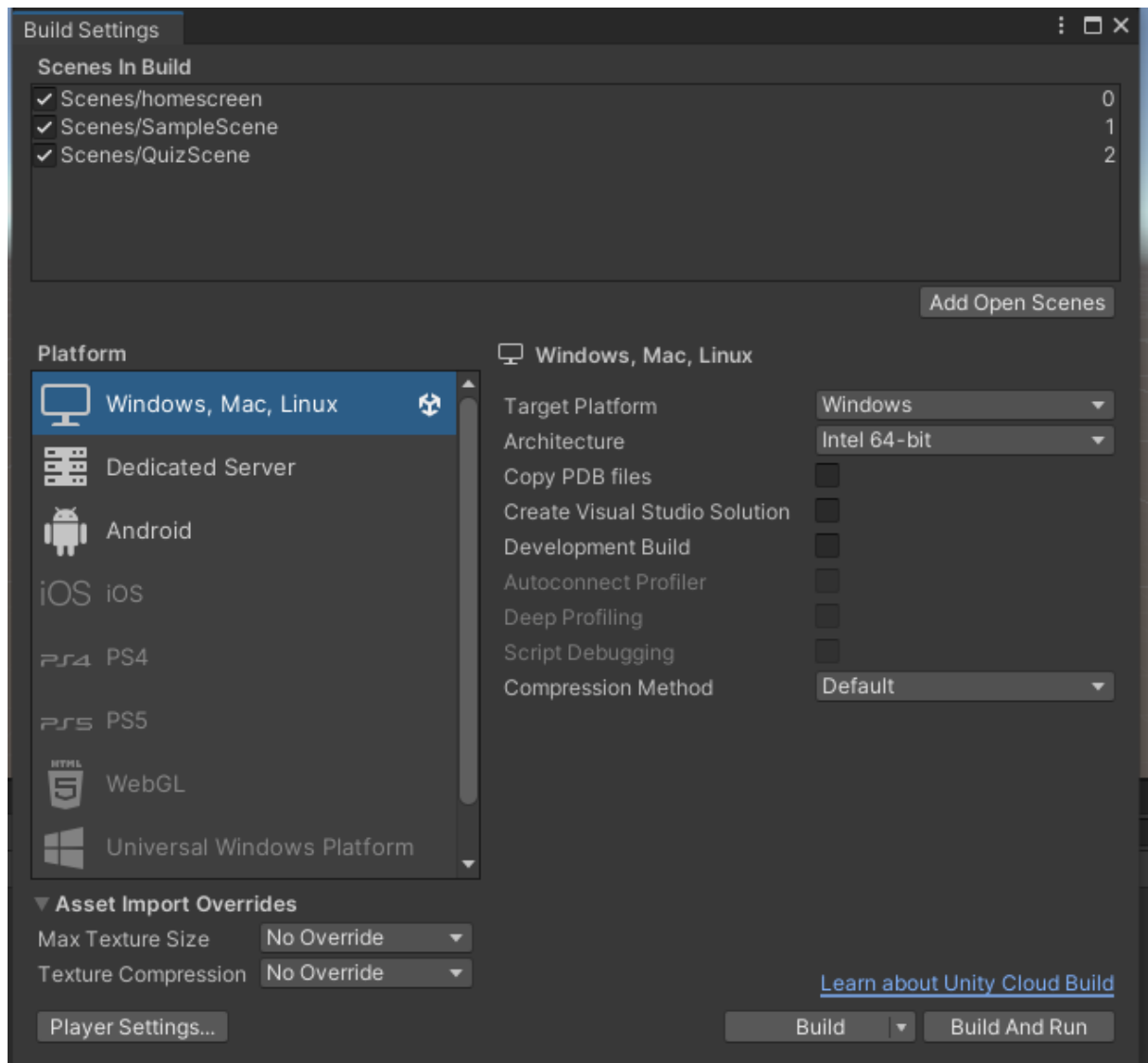
    // Metoda koja se poziva prije prvog ažuriranja kadra
    void Start()
    {
        // Sakrivanje tekstualnog panela i pokretanje rutine za prikaz teksta
        textPanel.SetActive(false);
        StartCoroutine(DisplayTextRoutine());
    }

    // Rutina za prikaz teksta
    IEnumerator DisplayTextRoutine()
    {
        // Prikazati tekstualni panel, pričekati određeno vrijeme, a zatim ga sakriti
        textPanel.SetActive(true);
        yield return new WaitForSeconds(displayDuration);
        textPanel.SetActive(false);
    }
}
```

Kod 10. Skripta za početni tekst koji nestaje

Izvor: autor

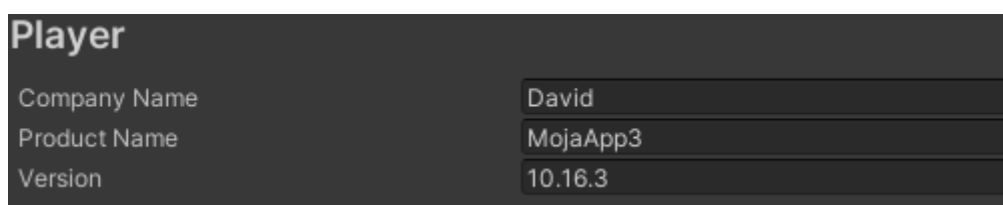
Sada je aplikacija spremna za korištenje. Još ju je samo potrebno kreirati. Otiđe se na *File* → *Build Settings* i klikne na *Build*.



Slika 27. Kreiranje aplikacije

Izvor: autor

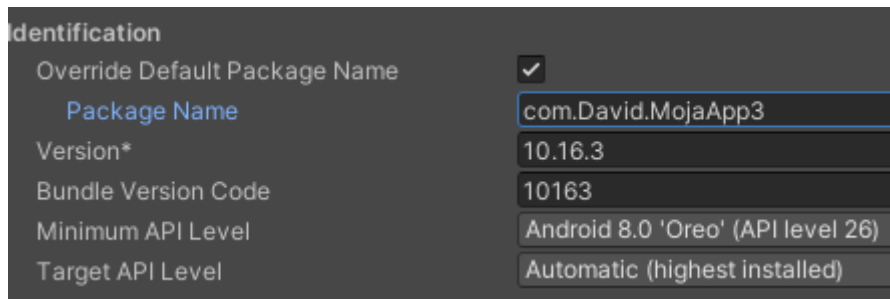
Aplikacija je sada spremna za korištenje na računalu. Da bi proradila na mobilnome uređaju, odnosno androidu na kojem će se aplikacija koristiti, potrebno je kliknuti na android i kliknuti na *player settings* te u odjeljku *player* podesiti sljedeće:



Slika 28. Postavke za Player Settings

Izvor: autor

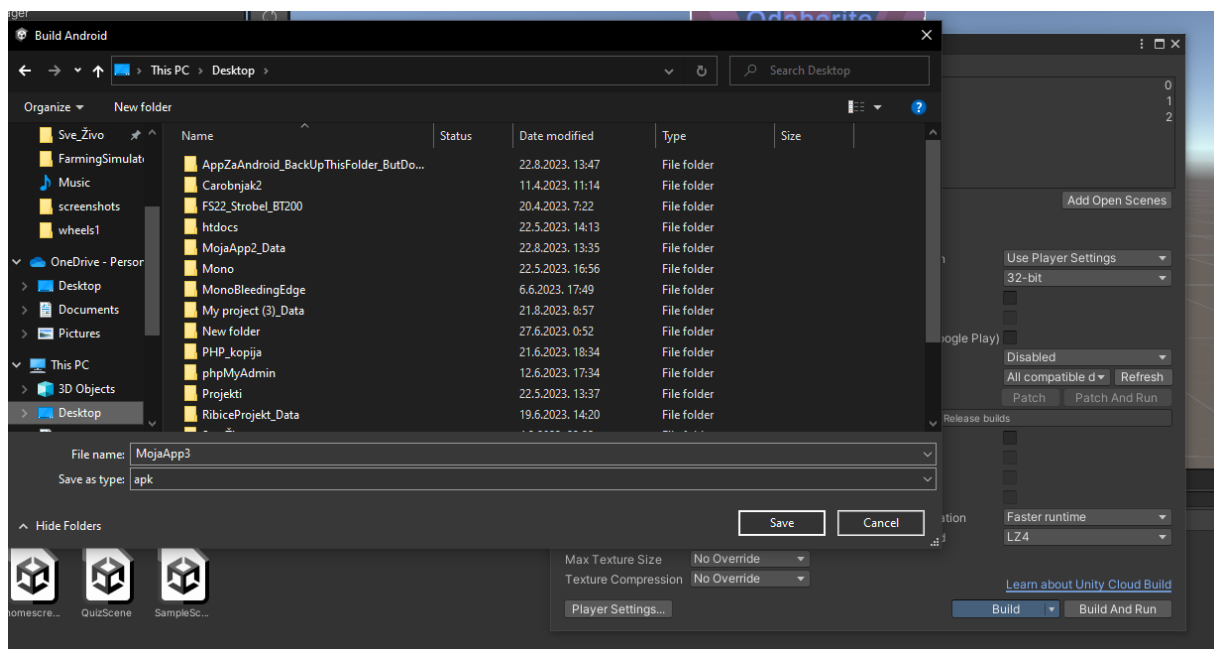
Pod *Company Name* upiše se ime tvrtke ili ime razvojnog inženjera (eng. *developer*) te se pod *Product Name* upiše ime projekta. Sada je još potrebno u odjeljku *Other Settings* pod *Identification* staviti *com.CompanyName.ProductName* kako je prikazano na slici 29.



Slika 29. Postavke za Identification

Izvor: autor

Nakon toga klikne se na *Build* i Unity će izraditi *apk* aplikaciju koja će se moći staviti na mobilni uređaj i koristiti na njemu.



Slika 30. Spremanje (engl. export) aplikacije

Izvor: autor

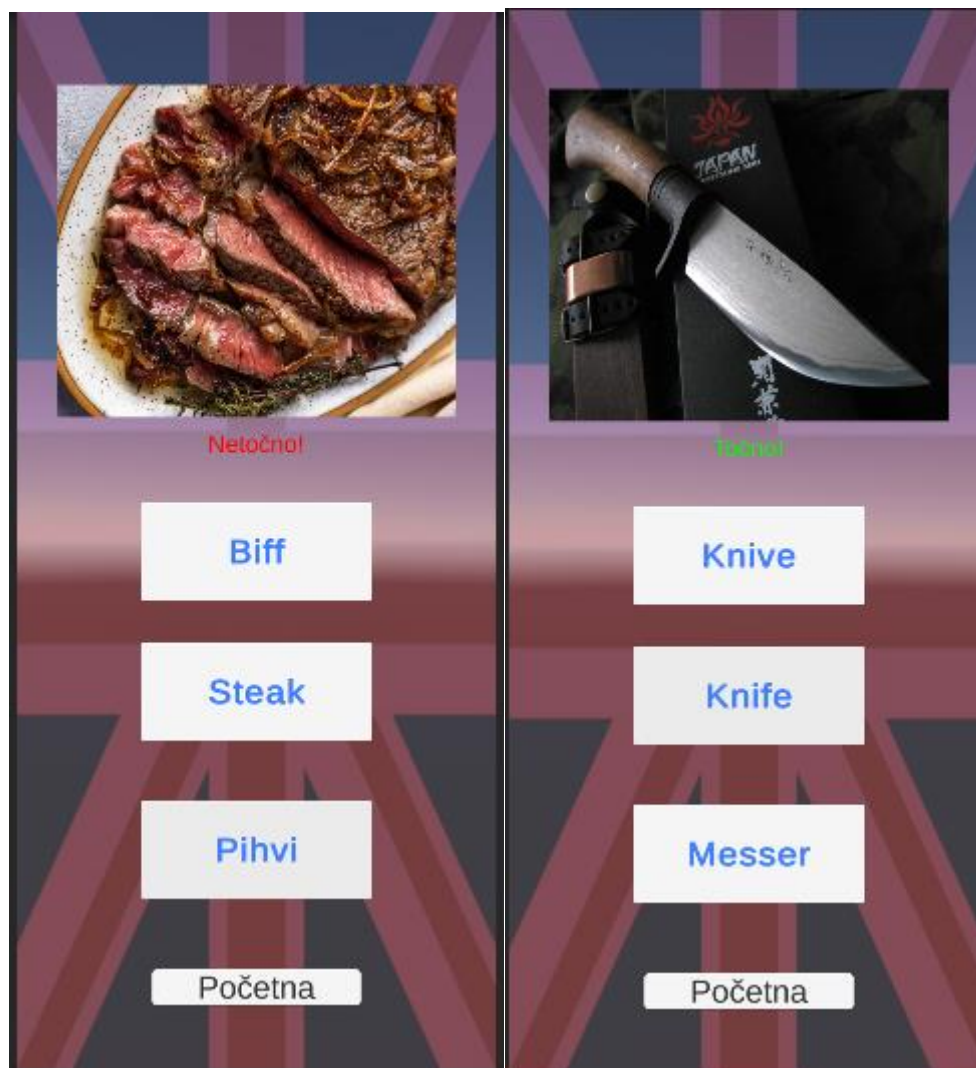
4. POSTIGNUTI REZULTATI

Rezultati se mogu vidjeti na slikama 31 i 32.



Slika 31. Prikaz početnog zaslona (lijevo) te prikaz rada AR kamere (desno)

Izvor: autor



Slika 32. Prikaz rada kviza

Izvor : Autor

Prikazivanjem objekata na slikama nije došlo ni do kakvih prepreka ili problema, a čak su i objekti s nižim *ratingom* iznjedrili zadovoljavajuće rezultate pri demonstraciji. Iako je to zahtijevalo pomak kamere bliže samoj slici, samim postupkom prikaza demonstrirana je visoka učinkovitost. Objekti su se zadržavali na slici čak i ako se kamerom nije pokrila cijela slika. Zvuk je bio besprijekoran, odmah se aktivirao tijekom prezentacije objekta i identifikacije slike. Nadalje, nije primijećeno nikakvo narušavanje glatkoće procesa ni kod provođenja provjere znanja ni kod završne scene, odnosno bodovanja.



Slika 33. Prikaz rada proširene stvarnosti

Izvor: autor



Slika 34. Završna scena s bodovima

Izvor: autor

5. KRITIČKI OSVRT

Sveukupno, ovaj rad kombinira tehničku dubinu s teorijskom osnovom pružajući čitatelju cjelovit uvid u proces izrade prototipne aplikacije za učenje jezika korištenjem tehnologije proširene stvarnosti. Neke od poteškoća koje su se javljale bile su stalno izgovaranje zvučnih datoteka pri pokretanju aplikacije, ali to je riješeno gašenjem funkcije *Play on Awake*. Naravno, aplikacija nije savršena te ima prostora za napredak. Moguće je dodavati kuboidne objekte, 3D cilindre, jednostavne 3D objekte te na taj način proširiti aplikaciju i na druge mogućnosti osim 2D slika koje su korištene u ovome radu.

Nadalje, također se ističe važnost kontinuirane interakcije i povratnih informacija korisnika kako bi se dalje unaprijedila funkcionalnost i korisničko iskustvo aplikacije. Pomoću takvih povratnih informacija moguće je identificirati dodatne poteškoće s kojima se korisnici susreću te implementirati poboljšanja da bi se korisničko iskustvo dovelo do što više razine udobnosti i praktičnosti.

Osim toga, buduće iteracije ove aplikacije mogle bi razmotriti integraciju naprednih tehnika obrade jezika što bi omogućilo razumijevanje konteksta i suptilnih nijansi učenja jezika. Također, dodavanje mogućnosti personalizacije, praćenja napretka korisnika i prilagodbe tečaja prema njihovim individualnim potrebama dodatno bi obogatilo aplikaciju i proširilo njezinu primjenu na širok spektar korisnika.

Ovaj rad ne samo da istražuje izradu prototipne aplikacije za učenje jezika tehnologijom proširene stvarnosti nego i naglašava kontinuiranu evoluciju i doradu takve aplikacije kako bi se zadovoljile potrebe korisnika. Kombinirajući tehničku dubinu s teorijskom podlogom, aplikacija ima potencijal transformirati način na koji usvajamo jezike inovativnim i interaktivnim metodama učenja. U budućnosti se očekuje da će takve tehnološki potpomognute obrazovne alatke postati još značajnije te bi mogle oblikovati budućnost učenja jezika i komunikacije.

6. ZAKLJUČAK

Ovaj završni rad istražuje primjenu tehnologije proširene stvarnosti (AR) u izradi prototipne aplikacije za učenje engleskog jezika koristeći se razvojnim okruženjem Unity i bibliotekama Vuforia. Cilj rada bio je olakšati proces učenja engleskog jezika interakcijom s virtualnim objektima u stvarnom vremenu te potaknuti aktivan angažman korisnika simuliranim scenarijima i zadacima.

Naglašena je važnost znanja stranih jezika u globaliziranom društvu, posebno engleskog jezika, kao i uloga tehnologije proširene stvarnosti kao inovativnog alata u obrazovanju. Postavljeni su ciljevi istraživanja i izrade aplikacije uključujući razvoj funkcionalnog prototipa koji omogućuje korisnicima pozitivno i korisno iskustvo učenja engleskog jezika.

Izradom prototipa korištenjem platforme Unity i biblioteke Vuforia ostvarena je interakcija s virtualnim objektima putem mobilnih uređaja. Prototip omogućuje korisnicima istraživanje, interakciju i praktično učenje engleskog jezika simuliranim scenarijima. Analiza postignutih rezultata potvrdila je prednosti aplikacije, uz istodobno prepoznavanje izazova i mogućih poboljšanja.

Kritička analiza primijenjenih metoda i postupaka ukazuje na pozitivne aspekte pristupa te otvara put za razmatranje alternativnih strategija za budući razvoj aplikacije. Ovaj rad pruža doprinos području obrazovanja i primjene tehnologije proširene stvarnosti potičući daljnja istraživanja i razvoj AR aplikacija u edukacijskom kontekstu.

Zaključno, rad je ukazao na potencijal tehnologije proširene stvarnosti za obogaćivanje procesa učenja jezika. Daljnji rad na razvoju takvih aplikacija može dodatno unaprijediti način na koji se jezici uče i potaknuti inovacije u edukacijskom sektoru. Ovaj rad pruža osnovu za daljnje istraživanje i implementaciju AR tehnologije u obrazovanju.

POPIS LITERATURE

- [1] Maryville University. Augmented Reality in Education: Interactive Classrooms: [Online]. 2023. Dostupno na: <https://online.maryville.edu/blog/augmented-reality-in-education/> (3.7.2023.)
- [2] Information Technology and Innovation Foundation. The Promise of Immersive Learning, Augmented and Virtual Reality's Potential in Education: [Online]. 2021. Dostupno na: <https://itif.org/publications/2021/08/30/promise-immersive-learning-augmented-and-virtual-reality-potential/> (3.7.2023.)
- [3] eLearning Industry. Augmented Reality (AR) In Education: [Online]. 2021. Dostupno na: <https://elearningindustry.com/augmented-reality-in-education-staggering-insight-into-future> (4.7.2023.)
- [4] Studica. Exploring Vuforia Augmented Reality Solutions for Education: [Online]. 2021. Dostupno na: [Exploring Vuforia Augmented Reality Solutions for Education - Studica Blog](#) (4.7.2023.)
- [5] MDPI. The effects of AR-enhanced English Language Learning Experience on Students' Attitudes, Self-Efficacy and Motivation: [Online]. 2021. Dostupno na: <https://www.mdpi.com/2079-9292/12/3/749> (4.7.2023.)
- [6] ViewSonic. The Benefits of AR in Education: [Online]. 2019. Dostupno na: <https://www.viewsonic.com/library/education/6-benefits-and-5-examples-of-augmented-reality-in-education/> (5.7.2023.)
- [7] School of education. Virtual Reality in Education: Benefits, Tools, and Resources: [Online]. 2019. Dostupno na: <https://soeonline.american.edu/blog/benefits-of-virtual-reality-in-education/> (10.7.2023.)
- [8] Logrocket. How to create an AR experience with Unity and Vuforia: [Online]. 2022. Dostupno na: <https://blog.logrocket.com/create-ar-experience-unity-vuforia/> (17.7.2023.)
- [9] Vuforia Library – PTC. Getting Started with Vuforia Engine in Unity: [Online]. 2023. Dostupno na: <https://library.vuforia.com/getting-started/getting-started-vuforia-engine-unity> (20.7.2023.)
- [10] Unity.com. Unity Engine: [Online]. 2019. Dostupno na: [Real-Time 3D Development Platform & Editor | Unity](#) (17.7.2023.)

- [11] Vuforia Engine Developer Portal. Downloads, Develop. License Menager, Target Menager: [Online]. 2023. Dostupno na: <https://developer.vuforia.com/downloads/sdk>, <https://developer.vuforia.com/> (21.7.2023.)
- [12] Microsoft. Dotnet. Microsoft: [Online]. 2023. Dostupno na: <https://dotnet.microsoft.com/en-us/languages/csharp> (22.7.2023.)
- [13] Microsoft. Visual studio: [Online]. 2023. Dostupno na: [Visual Studio: IDE and Code Editor for Software Developers and Teams \(microsoft.com\)](https://visualstudio.microsoft.com/) (22.7.2023.)
- [14] Unity Learn. Learn game development w/ Unity: [Online]. 2023. Dostupno na: <https://learn.unity.com/> (17.7.2023.)
- [15] Unity Documentation. Unity Manual: [Online]. 2023. Dostupno na: <https://docs.unity3d.com/Manual/UnityManual.html> (17.7.2023.)
- [16] Sound of Text: [Online]. 2013. Dostupno na: <https://soundoftext.com/> (2.8.2023.)

POPIS SLIKA

Slika 1. Izrada baze	9
Slika 2. Dodavanje <i>Image Targeta</i> u bazu.....	10
Slika 3. Prikaz svih <i>Image Targeta</i> (markera);.....	10
Slika 4. Prepoznatljivost slike;	11
Slika 5. Dodavanje markera u Unityju;	12
Slika 6. Dodavanje markera u Unityju	12
Slika 7. Prikaz hijerarhije nakon dodavanje markera (<i>Targeta</i>).....	13
Slika 8. Izgled hijerarhije nakon dodavanje teksta i objekata	13
Slika 9. Prikaz objekata u sceni.....	14
Slika 10. Prikaz objekata u sceni.....	14
Slika 11. Prikaz rada aplikacije u Unityju	15
Slika 12. Prikaz rada aplikacije u Unityju.....	15
Slika 13. Prikaz rada aplikacije u Unityju.....	16
Slika 14. Dodavanje zvuka za tanjur (engl. <i>plate</i>)	17
Slika 15. Prikaz dvaju polja napisanih u skripti	19
Slika 16. Prikaz postavki Canvasa	19
Slika 17. Izgled panela u obliku engleske zastave	20
Slika 18. Dodavanje slike na panel	21
Slika 19. Slika na panelu	22
Slika 20. <i>Level Menager</i>	22
Slika 21. Završni izgled početnog zaslona	24
Slika 22. Izgled zaslona koji predstavlja provjeru znanja	26
Slika 23. Postupak stvaranja klase <i>QuizData</i>	28
Slika 24. Dodavanje pitanja, odgovora i slika.....	29
Slika 25. Izgled scene za prikaz bodova.....	32
Slika 26. Tekst koji nestaje nakon tri sekunde	34
Slika 27. Kreiranje aplikacije	36
Slika 28. Postavke za <i>Player Settings</i>	36
Slika 29. Postavke za <i>Idenfication</i>	37
Slika 30. Spremanje (engl. <i>export</i>) aplikacije	37
Slika 31. Prikaz početnog zaslona (lijevo) te prikaz rada AR kamere (desno)	38
Slika 32. Prikaz rada kviza	39

Slika 33. Prikaz rada proširene stvarnosti	40
Slika 34. Završna scena s bodovima	41

POPIS KODOVA

Kod 1. Prikaz prvog dijela skripte koja predstavlja aktivaciju zvuka	18
Kod 2. Prikaz drugog dijela skripte koja predstavlja aktivaciju zvuka	18
Kod 3. <i>Level Menager</i>	23
Kod 4. <i>Scriptable Object</i>	27
Kod 5. Prvi dio skripte za upravljanje podacima iz klase <i>QuizData</i>	30
Kod 6. Drugi dio skripte za upravljanje podacima iz klase <i>QuizData</i>	30
Kod 7. Treći dio skripte za upravljanje podacima iz klase <i>QuizData</i>	31
Kod 8. Četvrti dio skripte za upravljanje podacima iz klase <i>QuizData</i>	31
Kod 9. Skripta za prikaz bodova	33
Kod 10. Skripta za početni tekst koji nestaje	35