Prilagodljiva arkadna igra u proširenoj stvarnosti

Matić, Krešimir

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Diplomski sveučilišni studij Računarstvo

PRILAGODLJIVA ARKADNA IGRA U PROŠIRENOJ STVARNOSTI

Diplomski rad

Krešimir Matić

Osijek, 2024.

SADRŽAJ

1. UVOD

Cilj ovog diplomskog rada je razviti mobilnu aplikaciju u proširenoj stvarnosti putem koje će biti moguće stvaranje lika na detektiranoj površini te pomoću kontrolera omogućiti liku da se kreće kroz stvarni svijet, tj. kroz kameru mobilnog uređaja. Tehnologija korištena za izradu ove aplikacije je Unity Game Engine koji pruža mnoštvo vanjskih alata za izradu igara u virtualnoj i proširenoj stvarnosti poput *XR Interaction Toolkita* i *AR Foundationa*. Pomoću ovih alata implementirat će se mogućnost detekcije površina kroz kameru mobilnog uređaja te sve potrebne interakcije koje će korisnik koristiti u ovoj aplikaciji.

U ovom diplomskom radu detaljno će se obraditi tematika primjene proširene stvarnosti, kako u svijetu video igara, tako i u marketingu i drugim aspektima svakodnevnog života. Proučavat će se temeljni koncepti proširene stvarnosti, uključujući njezine prednosti, nedostatke i različite primjene. Rad će također pružiti detaljan uvid u korištenu tehnologiju, s posebnim naglaskom na Unity Game Engine i pomoćne alate koji su korišteni za ostvarivanje ciljeva ovog projekta. U glavnom dijelu rada biti će objašnjen postupak korištenja navedenih tehnologija u svrhu izrade aplikacije sa ispunjenim svim uvjetima rada. Objasnit će se svaki aspekt i funkcionalnost aplikacije poput detekcije površine, stvaranje lika na detektiranu površinu te kretanje lika kroz proširenu stvarnost. Za potrebe stvaranja lika potrebno je imati i odgovarajući model koji se može pronaći na raznim web stranicama za Unity modele. Uz model lika potrebno je stvoriti i odgovarajuće animacije kretanja lika zajedno sa animacijama okoline. Također će se objasniti mehanika konačnog cilja igre te na koji način je izvedena.

1.1. Zadatak diplomskog rada

U zadatku je potrebno opisati način stvaranja igara u proširenoj stvarnosti primjenom *Unitya*. Zatim je potrebno napraviti igru koja će se moći igrati na bilo kojim zaprekama, tj. na različitim terenima u stvarnome svijetu.

2. PROBLEMATIKA I RJEŠENJA U PROŠIRENOJ STVARNOSTI

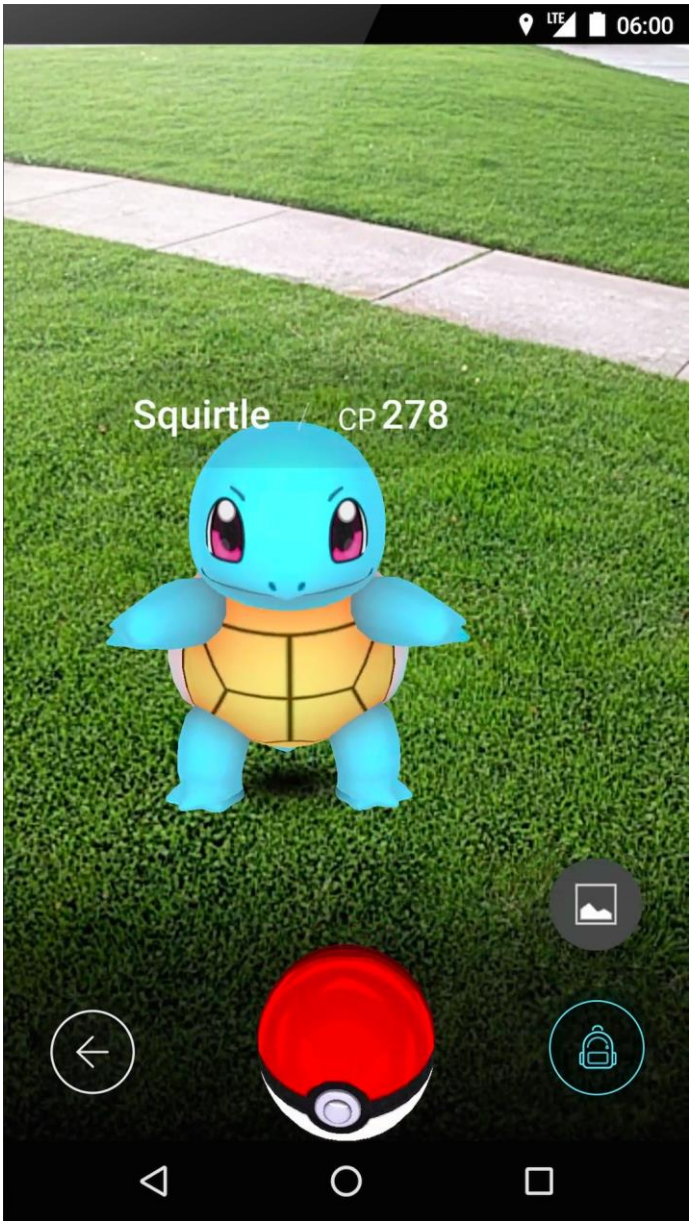
Razvoj aplikacija u proširenoj stvarnosti, dalje u tekstu AR (*engl. Augmented reality*) je izazovan zadatak. Programer se mora brinuti o raznim stvarnim, prvotno o detekciji površina kroz kameru uređaja kako bi se omogućile daljnje funkcije te stvaranje scene u proširenoj stvarnosti. Postoje mnogi primjeri igara u proširenoj stvarnosti poput: Pokemon GO, Harry Potter: Wizards Unite, Ingress, Jurassic World Alive, Minecraft Earth i mnoge druge. No, primjena proširene stvarnosti nije samo u industriji video igara, već postoje mnoge primjene ove tehnologije u velikom broju industrija poput medicine, industrijske proizvodnje, školstvu itd. Posebno primjena u medicini je od velike važnosti gdje studenti i doktori mogu primijeniti svoja znanja i vještine kroz operaciju u proširenoj stvarnosti u svrhu usavršavanja svoje struke. Kako u medicini tako i u školstvu, kroz proširenu stvarnost učenici mogu vizualizirati određene objekte u svrhu boljeg razumijevanja lekcija i zadataka.

2.1. Pokémon GO

Pokémon GO, igru koju je razvio studio *Niantic* zajedno u suradnji s *Nintendom* i *The Pokémon Company*, je mobilna igra proširene stvarnosti koja je objavljena u srpnju 2016. Igra je brzo postala globalni fenomen, koristeći AR tehnologiju za spajanje omiljenog *Pokémon* svemira sa stvarnim svijetom. Igrači koriste svoje pametne telefone za traženje i snimanje virtualnih *Pokémona*, koji se preko kamere na telefonu pojavljuju u stvarnom svijetu. Inovativna upotreba AR-a u igri stvara impresivno iskustvo u kojem se igrači osjećaju kao pravi *Pokémon* treneri, istražujući svoja susjedstva, parkove i gradove u potrazi za *Pokémonima* koje će uhvatiti.

Tehnički gledano, *Pokémon GO* koristi kombinaciju GPS-a i tehnologije kamere kako bi pružio svoje AR iskustvo. Igra koristi GPS podatke za mapiranje igračeve lokacije u stvarnom svijetu i prekrivanje s virtualnom kartom igre koja uključuje *PokéStopove*, teretane i točke za stvaranje *Pokémona*. Virtualna karta se kontinuirano ažurira na temelju kretanja igrača, osiguravajući da je igranje uvijek relevantno za trenutnu lokaciju igrača. *Nianticova* platforma *Real World* obrađuje ove podatke o lokaciji, integrirajući ih s vizualnim elementima proširene stvarnosti i mehanikom igre [1].

AR funkcionalnost primarno pokreću kamera i žiroskop pametnog telefona, koji zajedno omogućuju igri da postavi 3D modele *Pokémona* na stvarni prikaz svijeta koji se vidi kroz kameru. Ova kombinacija hardvera omogućuje *Pokémonima* da izgledaju kao da postoje u okruženju igrača, stvarajući besprijekoran spoj virtualnog i stvarnog svijeta. Igra koristi napredne algoritme za prepoznavanje slike i praćenje pokreta kako bi održala položaj i veličinu *Pokémona* u odnosu na točku gledišta kamere, povećavajući realizam AR iskustva. Na Slika 1 je prikazana scena u igri [1].

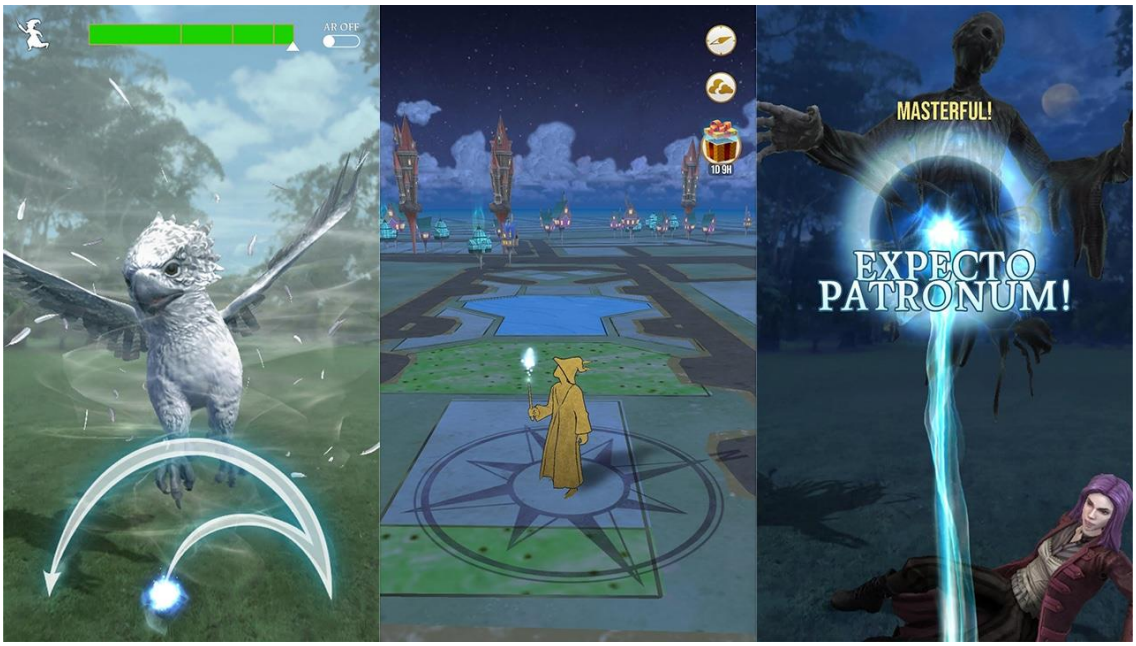


Slika 1. Prikaz scene *Pokémon GO* aplikacije [2].

2.2. Harry Potter: Wizards Unite

Mobilna igra u proširenoj stvarnosti, koju je također razvio studio *Niantic* sa suradnjom s *WB Games San Francisco* i *Portkey Games, Harry Potter: Wizards Unite* je uranjala igrače u čarobni svijet Harry Pottera, koristeći AR tehnologije za spajanje čarobnjačkog svemira sa stvarnim svijetom dok joj nažalost nije došao kraj u siječnju 2022. Igrači su koristili svoje mobilne uređaje da istražuju svoje okruženje, bacaju čarolije, otkrivaju čarobne artefakte i susreću magična stvorenja i likove iz serijala *Harry Potter* i *Fantastic Beasts*. Inovativna upotreba proširene stvarnosti u igri stvarala je iskustvo u kojem se igrači osjećaju poput čarobnjaka i vještica, istražujući svoje četvrti, parkove i gradove u svrhu čarobnih susreta.

Gledajući sa tehničke strane, poput *Pokémon GO* igre, *Harry Potter: Wizards Unite* također je koristio kombinaciju GPS-a i tehnologija kamere u svrhu pružanja AR iskustva. GPS podaci su se koristili za mapiranje lokacije igrača u stvarnom svijetu i njihovo preklapanje s virtualnom kartom koja je sadržavala gostionice, staklenike, tvrđave i točke mrjestilišta za čarobne susrete zvane "*Foundables*". Virtualna karta se neprestano osvježavala prateći kretanje igrača, čime se osiguravalo da je igra uvijek usklađena s trenutnom lokacijom igrača. Kao i u *Pokémon GO,* koristila se platforma *Real World* za obradu podataka o lokaciji. Slika 2 prikazuje tri scene iz igre [3].



Slika 2. Prikaz tri scene mobilne igre *Harry Potter: Wizards Unite* [3].

2.3. Jurassic World Alive

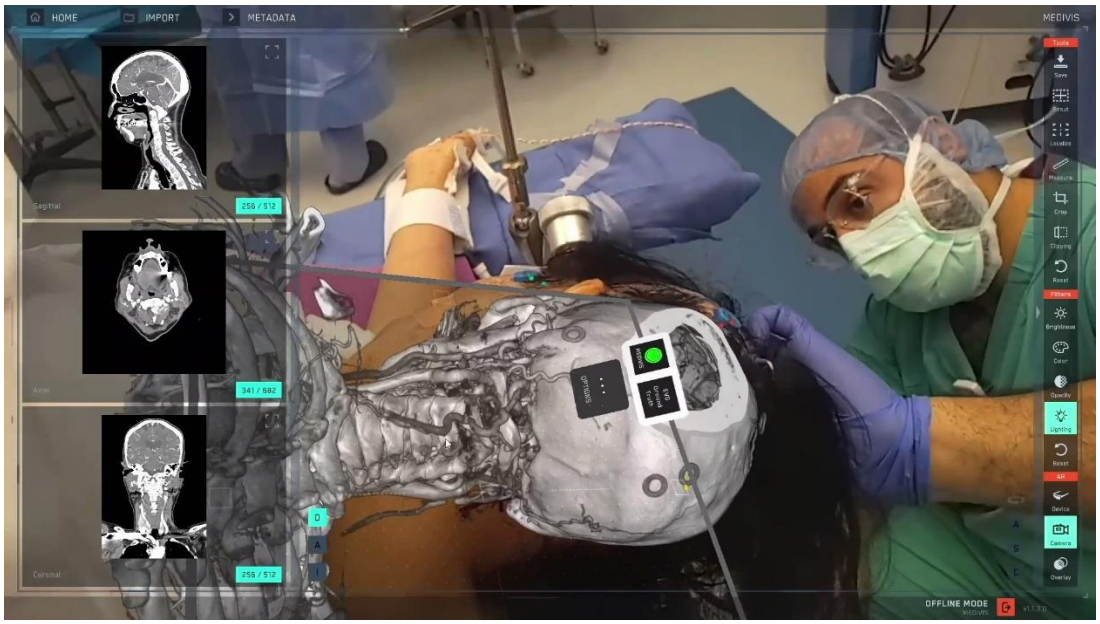
Jurassic World Alive je mobilna igra u proširenoj stvarnosti objavljena u svibnju 2018. od strane *Ludiae*. Igra uranja igrače u svijet u kojem dinosauri lutaju modernom Zemljom. Sličnim konceptom kao u prethodnim primjerima, igrači koriste svoje mobilne uređaje da istražuju svoje okruženje, no u ovom slučaju otkrivaju i skupljaju dinosaure, koji se preko kamere pojavljuju u stvarnom svijetu. Korištenje tehnologije proširene stvarnosti unutar igre, stvara se iskustvo gdje se igrači osjećaju kao pravi treneri dinosaura, istražujući okolinu kako bi pronašli i prikupili uzorke DNK različitih vrsta dinosaura. I u ovoj igri se koristi GPS zajedno za kamerom da bi se pružilo samostalno AR iskustvo. Kao u primjeru *Pokémon GO* igre, postoje određene točke na karti koje se pomoću GPS-a mapiraju te na kojima se mogu pronaći dinosauri i opskrbne točke. Igra uključuje različite elemente mobilne mrežne tehnologije. Kontinuirana razmjena podataka između klijenta i *Ludijinih* poslužitelja osigurava sinkronizaciju podataka igre u stvarnom vremenu, poput lokacije igrača i susreta s dinosaurima [4]. Detaljan prikaz igre može se vidjeti na Slika 3.

Slika 3. Prikaz tri scene iz igre *Jurassic World Alive* [4].

2.4. SurgicalAR

SurgicalAR, koju je razvio studio *Medivis,* koristi AR tehnologiju za pomoć kirurzima tijekom operaicja pružajući 3D vizualizaciju anatomije pacijenta izravno postavljanu na mjesto operacije. Korištenjem uređaja *Microsoft HoloLens, SurgicalAR* omogućuje kirurzima vizualizaciju kritičnih struktura kao što su krvne žile, kosti i tumori u stvarnom vremenu, povećavajući preciznost i poboljšavajući kirurške ishode. Neki od ključnih značajki ove aplikacije su predoperativno planiranje gdje kirurzi mogu koristiti detaljne 3D modele pacijentove anatomije izvedene iz CT i MRI skeniranja za planiranje kirurških zahvata, intraoperativno vođenje gdje tijekom operacije AR sustav projicira navedene 3D modele na pacijentovo tijelo pružajući smjernice u stvarnom vremenu [5].

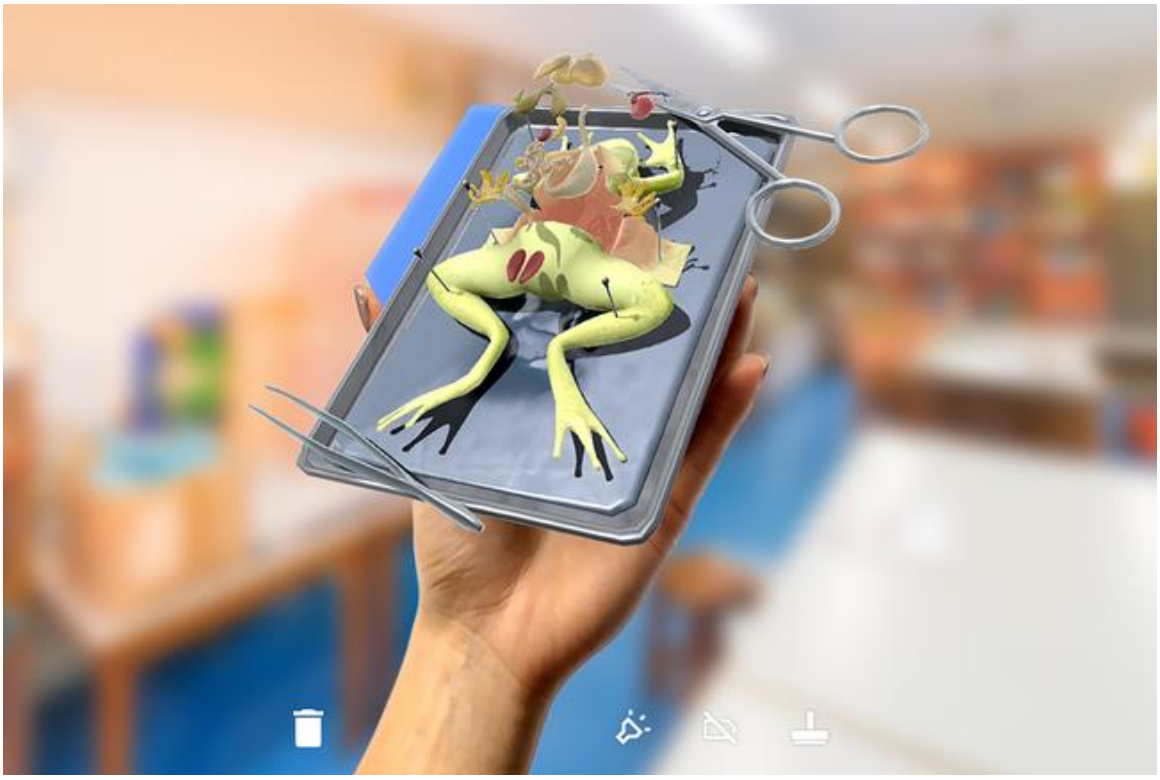
Jedna od svrha ove aplikacije je poboljšanje sposobnosti kirurga da vidi kritičnu anatomiju te na taj način smanjiti rizik od komplikacija i poboljšati točnost kirurških intervencija. Također, može služiti kao obrazovni alat koji studentima medicine omogućuje promatranje složenih postupaka s detaljnim anatomskim kontekstom koji poboljšava učenje i razumijevanje. Slika 4 prikazuje ovaj alat u upotrebi [5].



Slika 4. Prikaz *SurgicalAR* aplikacije [7].

2.5. Merge EDU

Merge EDU je platforma proširene stvarnosti osmišljena za poboljšanje iskustva učenja za studente pružanjem interaktivnog, 3D obrazovnog sadržaja. Platforma uključuje niz alata i resursa koji oživljavaju teme poput znanosti, matematike i povijesti kroz impresivna AR iskustva. Aplikacija nudi širok raspon AR sadržaja usklađenog s obrazovnim standardima. To uključuje detaljne 3D modele bioloških organizama, povijesnih artefakata, planeta i geometrijskih oblika. Korisnici mogu manipulirati ovim modelima kako bi bolje razumjeli predmet. Platforma nudi unaprijed dizajnirane AR lekcije i aktivnosti koje su osmišljene da budu interaktivne i privlačne, pomažući da korisnici lakše zapamte informacije. Dizajnirana je da bude jednostavna za korištenje i dostupna na različitim uređajima, uključujući tablete i mobilne uređaje. *Merge EDU* je primjer kako AR može transformirati tradicionalne obrazovne metode pružajući studentima impresivna, interaktivna iskustva učenja koja poboljšavaju razumijevanje različitih predmeta. Primjer rada aplikacije se može vidjeti na Slika 5 [6].



Slika 5. Prikaz *Merge EDU* aplikacije [6].

3. KORIŠTENE TEHNOLOGIJE I ALATI

Tehnologije korištene u svrhu izrade mobilne aplikacije u proširenoj stvarnosti za ovaj rad su prvobitno *Unity Game Engine.* Za razvoj aplikacija u proširenoj stvarnosti koristeći *Unity* potrebni su određeni dodaci, a to su *XR Interaction Toolkit* i *AR Foundation*. Oba alata su dostupna u *Unity* registru za jednostavnu integraciju unutar projekta.

3.1. Unity Game Engine

Unity je razvojni alat poznat po svojoj fleksibilnosti i sveobuhvatnom skupu alata. Stekao je popularnost među programerima u raznim industrijama kao što su igre, film, arhitektura i inženjerstvo. Nudi cjelovitu platformu za stvaranje 2D i 3D igara, interaktivnih iskustava i simulacija. Sveobuhvatan je u smislu podrške za mnoštvo platformi, uključujući mobilne, desktop, web, konzole i uređaje za virtualnu i proširenu stvarnost, tj. VR i AR uređaje. Te ovim pristupom gdje se nakon razvoja aplikacije s lakoćom može promijeniti platforma, uvelike smanjuje vrijeme i napor razvoja, omogućujući programerima da se koncentriraju na izradu zanimljivog sadržaja bez ograničenja specifične platforme. Kroz *Unity,* omogućeno je fizički temeljeno renderiranje, globalno osvjetljenje u stvarnom vremenu i sofisticirana opcija shader elemenata za osjenčavanje prostora, koje programerima omogućuje stvaranje okruženja sa zadivljujućim realizmom. Također je vrijedno spomenuti jednostavnost korištenja sučelja gdje su sve opcije intuitivne i lake za shvatiti. Moguće je i vlastoručno prilagođavanje izgleda sučelja ovisno o korisniku [8].

Zbog mogućnosti razvoja u stvarnom vremenu, *Unity* je postao popularan alat ne samo za razvoj video igara, već i u drugim industrijama poput filmske i automobilske industrije. U produkciji filma, koristi se za stvaranja animacija i vizualnih efekata, omogućavajući korisnicima vizualizaciju scena i pravljenje izmjena u trenu. U automobilskoj industriji koristi se za dizajn i vizualizaciju automobilskih prototipova, stvarajući zapanjujuće pokazne prostorije, i razvoj interaktivnih simulacija za treniranje. *Unity* se može primijeniti u bilo kojoj industriji koja zahtijeva visokorazinsku vizualizaciju i interaktivna iskustva, zbog sposobnosti rukovanja složenim simulacijama i interakcija u stvarnom vremenu [8].

3.2. XR Interaction Toolkit

XR Interaction Toolkit je interaktivni sustav visoke razine koji se temelji na komponentama za stvaranje iskustava u virtualnoj i proširenoj stvarnosti. Pruža okvir koji čini dostupnim 3D i UI interakcije iz *Unity* input događaja. Srž ovog sustava je skup osnovnih *Interactor* i *Interactable* komponenti, te upraviteljem interakcija koji povezuje ove dvije vrste komponenti. Također sadrži komponente koje se mogu koristiti za kretanje i crtanje vizualnih komponenti. *XR Interaction Toolkit* sadrži skup komponenti koje podržavaju sljedeće zadatke interakcije:

- Mogućnosti korištenja XR kontrolera sa više platformi: *Meta Quest, OpenXR, Windows Mixed Reality* itd.
- Osnovno lebdjenje iznad objekta, odabir i hvatanje objekta
- Haptička povratna informacija putem XR kontrolera, tj. prijenos informacija pomoću dodira ili fizičkog osjećaja
- Vizualna povratna informacija za označavanje mogućih i aktivnih interakcija
- Osnovna UI interakcija putem XR kontrolera
- Uslužni program za rad sa XR *Origin*om [9]

3.3. AR Foundation

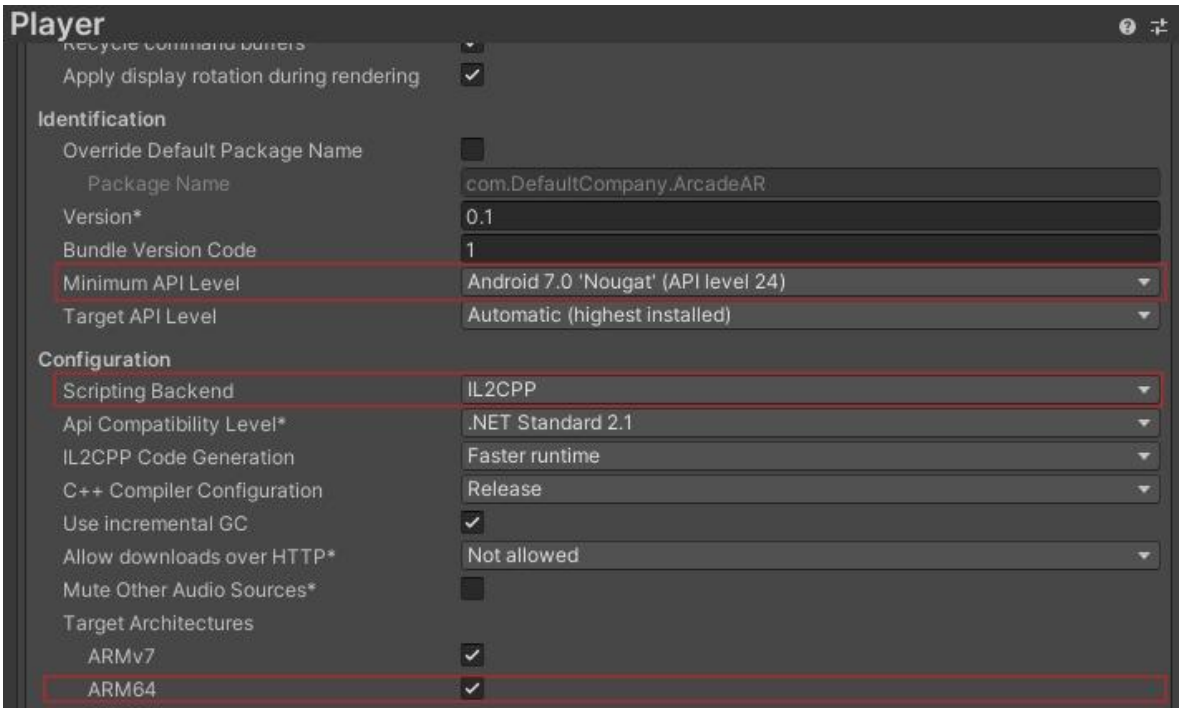
AR Foundation omogućuje stvaranje višeplatformskih aplikacija proširene stvarnosti s *Unityem*. Unutar *AR Foundation* projekta moguće je odabrati koje AR značajke se žele koristiti na način da se dodaju odgovarajuće upravljačke komponente unutar scene. Pomoću izvornog AR

SDK-a platforme, moguća je izrada i implementiranje aplikacije na vodeće svjetske AR platforme [10].

Slika 6. Primjer scene za detekciju površina i *raycast* unutar *AR Foundationa* [10].

4. RAZVOJ AR APLIKACIJE

Razvoj aplikacije počinje osnovnim kreiranjem *Unity3D* projekta. Kao što je spomenuto u prethodnom podnaslovu, za rad sa proširenom stvarnošću potrebno je unutar projekta dodati alate *XR Interaction Toolkit* i *AR Foundation* putem upravitelja projekta. Ovi alati su dostupni unutar *Unity* registra te nije potrebno dodatno preuzimanje sa vanjskih stranica. Nakon dodavanja svih potrebnih dodataka, potrebno je prebaciti projekt na odgovarajuću platformu, tj. na *Android Build.* Prilikom prebacivanja na odgovarajuću platformu, potrebno je i postaviti odgovarajuće postavke da bi se aplikacija uspješno mogla pokrenuti na svim mobilnim uređajima. U postavkama igrača unutar *Android* taba potrebno je postaviti minimalnu API razinu, u ovom slučaju postavljeno je na *Android 7.0 'Nougat'*, *backend* skriptiranje je potrebno prebaciti sa Mono na IL2CPP, te omogućiti ARM64 kao ciljanu arhitekturu uz ARMv7. Slika 7 prikazuje sve izvršene promjene označene crvenim okvirom. Sljedeći korak je brisanje zadane kamere u sceni te kreiranje novog objekta *XR Origin*, koji je dio *AR Foundation* alata. Ovaj objekt se ponaša kao roditeljski objekt izvedene glavne kamere te određuje poziciju korisnika aplikacije. Ova kamera se ponaša kao kamera mobilnog uređaja te kroz nju će korisnik vidjeti svijet proširene stvarnosti te stupiti u interakciju s njim.



Slika 7. Izmijenjene postavke u svrhu rada na *Android* uređajima.

Nakon postavljanja projekta moguće je krenuti sa razvojem aplikacije. Razvoj se može podijeliti u više dijelova, a to su:

- Detekcija površina
- Stvaranje lika na detektiranoj površini
- Omogućavanje kretanja lika kroz proširenu stvarnost
- Kreiranje jednostavnog sučelja za kretanje među scenama
- Cilj igre
- Zvučni efekti
- Stvaranje animacija za sve potrebne objekte

4.1. Detekcija površina

Kao prvi korak detekcije površine potrebno je stvoriti novi objekt unutar scene pod nazivom *XR Interaction Manager* čija je svrha obrada i ujedinjavanje svih interakcija u sceni. Zatim je potrebno unutar *XR Origin* objekta dodati komponentu pod nazivom *AR Plane Manager.* Unutar ove komponente moguće je postaviti na koji način će se detektirati površina, da li će se detektirati samo horizontalna površina, vertikalna površina ili u slučaju ove aplikacije oboje. Također je moguće postaviti izgled detektirane površine putem *prefaba.* To je moguće uraditi stvaranjem

novog objekta u scenu pod nazivom *AR Default Plane.* Ova zadana površina ima tri komponente od interesa, a to su *AR Plane* komponenta, *AR Plane Mesh Visualizer* i *Mesh Renderer*. *AR Plane* sadrži dvije opcije za prilagođavanje gdje je moguće osposobiti brisanje ako površina više nije detektirana te prilagođavati maksimalnu vrijednost pri kojoj položaj vrha površine se promijeni da pozove događaj promjene granica. *AR Plane Mesh Visualizer* je komponenta gdje se iz samog naziva može zaključiti da omogućava vizualizaciju detektirane površine. Unutar ove komponente je moguće promijeniti kvalitetu praćenja površine te je poželjno to uraditi sa opcije *Limited*, gdje je praćenje površina ograničeno ili slabe kvalitete, na opciju *Tracking* prema kojoj praćenje rade normalno [11]. Također, postoji opcija za onemogućavanje određenih dijelova komponente, tj. *mesh* i *line rendera*, u slučaju da se spoje i preklapaju dvije ili više detektiranih površina. Za *mesh renderer* komponentu najznačajnije je spomenuti da se putem nje može promijeniti materijal korišten za vizualizaciju detektirane površine jednostavnim dodavanjem prilagođenog materijala unutar taba za materijale. Kao referenca se može koristiti zadani materijal površine.

4.2. Stvaranje lika na detektiranoj površini

Nakon uspješne detekcije površine sljedeći korak je omogućavanje stvaranja lika jednostavnim pritiskom na površinu. Za ovu svrhu potrebno je dodati *AR Placement Interactable* u scenu, te potrebno je unutar *XR Origin* objekta dodati *AR Raycast Manager*. *AR Placement Interactable* nudi sve potrebne funkcionalnosti za stvaranje objekta na detektiranoj površini unutar proširene stvarnosti.. U ovom objektu potrebno je izbrisati istoimenu komponentu, tj. skriptu, te dodati novu skriptu koja *overridea* izbrisanu. Na Slika 8 prikazane su dodatne funkcionalnosti što nudi *override-ana* verzija *AR Placement Interactable* skripte. Zadržane su sve osnovne funkcionalnosti, no postavljeno je da rotacija objekta ovisi o kameri te stvoreni objekt se ne može okretati oko y-osi, tj. objekt ne može pasti na stranu [12].

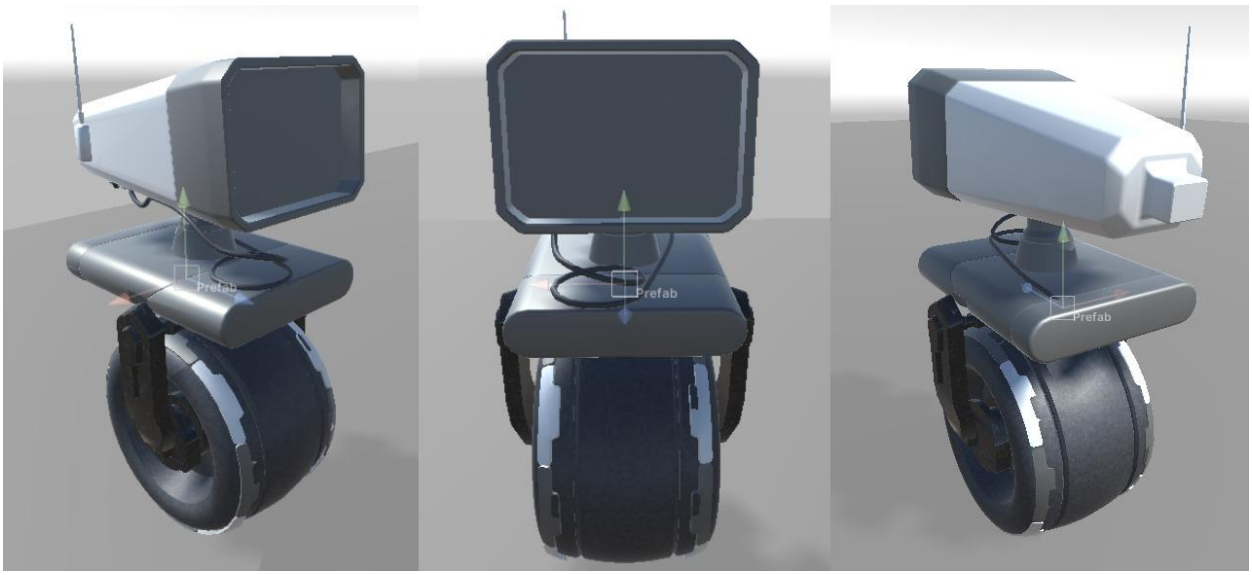
```
using System.Collections;
using System.Collections.Generic;
using Unity.XR.CoreUtils;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit.AR;

public class OverrideARPlacementInteractable : ARPlacementInteractable
{

    protected override void OnObjectPlaced(ARObjectPlacementEventArgs args)
    {
        base.OnObjectPlaced(args);
        Vector3 rotation = Camera.main.transform.position - args.placementObject.transform.position;
        rotation.y = 0;
        args.placementObject.transform.rotation = Quaternion.LookRotation(rotation);
        this.enabled = false;
    }
}
```

Slika 8. *Overrideana AR Placement Interactable* skripta

Prefab objekta koji će se stvoriti mora biti dodan unutar ove komponente pod *placement prefab* opciju. Zatim, u svrhu stvaranja samo jednog objekta na površinu, potrebno je nakon prvog stvaranja objekta, unutar interaktivnih događaja ove komponente onemogućiti samu komponentu za stvaranje. Sve interakcije ove komponente su omogućene pomoću *XR Interaction Managera* koji je posebno dodan, te je interakcijska maska postavljena na „sve“, što znači da je moguća interakcija sa svim interaktorima u sceni. S obzirom na *AR Raycast Manager,* on služi kao medijator interakcija korisnika sa scenom. Moguće je dodati *prefab* u svrhu vizualizacije zrake, ali to nije potrebno u ovoj aplikaciji zbog preglednosti konačnog rješenja.

4.3. Model lika

Kao što je spomenuto u prethodnom poglavlju, potrebno je postaviti *placement prefab* koji će se ponašati kao lik kojim će biti moguće upravljati. Tijekom testne faze stvaranja na površinu korištena je jednostavna kocka, no za svrhe ovog rada potrebno je kreirati ili pronaći zadovoljavajući model lika. Rješenje je pronađeno putem stranice *Sketchfab* [16]. To je stranica za objavljivanje, dijeljenje, otkrivanje, kupnju i prodaju 3D, VR i AR sadržaja. Na Slika 9 je prikazan model korišten za svrhe ovog rada. Preuzimanjem modela sa stranice dobije se *.zip* datoteka unutar koje se nalazi 3D model sa odgovarajućim slikama od kojih je potrebno stvoriti odgovarajuće materijale unutar *Unitya*. Prilikom uvoza modela u *Unity* potrebno je testirati određene aspekte modela, kao npr. orijentacija modela i skaliranje modela. Kada se testiraju i po potrebi poprave odgovarajuće značajke, i postave se svi potrebni materijali na model, moguće je krenuti na sljedeći korak, a to je omogućavanje upravljanja lika.



Slika 9. Prikaz modela lika

4.4. Omogućavanje kretanja lika kroz proširenu stvarnost

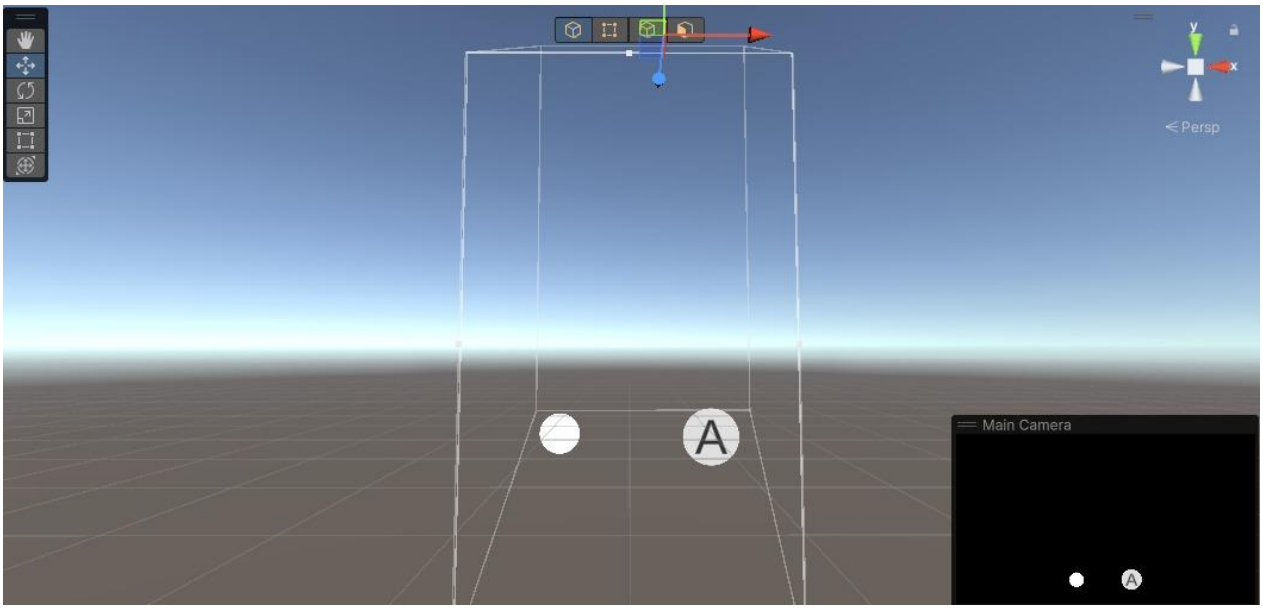
Nakon kreiranja lika na površini potrebno je osmisliti način kretanja po svim detektiranim površinama. Potrebno je omogućiti kretanje u svim smjerovima te dodati mogućnost skoka. Ovo je moguće uraditi kreiranjem nove skripte pod nazivom *PlayerController.cs* te dodavanjem je na *prefab* lika. UI za kretanje se sastoji od *joystick* kontrolera te gumba za skakanje na suprotnim stranama ekrana. *Joystick* kontroler je preuzet sa *Unity Asset Storea.* Način kretanja je osmišljen da pomjeranjem *joystick* kontrolera, računaju se horizontalne i vertikalne vrijednosti te se dodaju na *rigidbody.velocity* komponentu lika. Uz ovo računanje osmišljen je i način da je kretanje lika uvijek ovisno o poziciji glavne kamere u slučaju da se mijenjanjem pozicije kamere u stvarnom vremenu ne dođe do nepravilnih kontrola. Unutar ove skripte također postoji metoda za skakanje gdje se na *rigidbody* komponentu poziva metoda *AddForce()* sa prilagodljivom snagom skoka. No, postoji uvjet gdje se provjerava da li je lik prizemljen te u tom slučaju je tek moguće skakanje. Ta provjera se vrši na način da iz središnjeg dijela lika, šalje se *raycast* prema tlu te ako lik dodiruje tlo uvjet za skakanje je ispunjen. Slika 10 prikazuje *PlayerController.cs* skriptu.

```
[RequireComponent(typeof(Rigidbody))]
public class PlayerController : MonoBehaviour
{
    [SerializeField] private Rigidbody rb;
    [SerializeField] private CapsuleCollider capsuleCollider;
    [SerializeField] private LayerMask layerMask;
    [SerializeField] private float movementSpeed = 1;
    [SerializeField] private float jumpForce = 5f;

    private FixedJoystick joystick;

    private bool isGrounded;

    private void Awake()
    {
        joystick = FixedJoystick.Instance;
    }
    private void FixedUpdate()
    {
        rb.velocity = (joystick.Horizontal * movementSpeed * Vector3.ProjectOnPlane(Camera.main.transform.right, Vector3.up)) +
            (joystick.Vertical * movementSpeed * Vector3.ProjectOnPlane(Camera.main.transform.forward, Vector3.up));
        isGrounded = Physics.Raycast(rb.worldCenterOfMass, -Vector3.up, (capsuleCollider.height / 2) * 1.01f * rb.transform.localScale.y, layerMask);

        if (joystick.Horizontal != 0 ||  joystick.Vertical != 0)
        {
            transform.rotation = Quaternion.LookRotation(rb.velocity);
        }
    }

    public void Jump()
    {
        if (isGrounded)
        {
            rb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
        }

    }
}
```

Slika 10. Prikaz *PlayerController.cs* skrpte.

Sljedeći korak je kreiranje UI-a na koji će biti smješteni *joystick* komponenta i gumb za skakanje. U tu svrhu potrebno je kreirati *Canvas* te postaviti unutar komponente način renderiranja na opciju *Screen Space – Camera* te unutar praznog polja *Render Camera* dodati glavnu kameru iz *XR Origin* komponente. Također je potrebno postaviti skaliranje sa veličinom ekrana te referentnu rezoluciju postaviti na 1080 x 1920 piksela. Zatim je potrebno dodati *Fixed Joystick* i gumb komponentu na *Canvas* te prilagoditi im izgled i pozicije u svrhu preglednosti. Izgled UI-a, tj. poredak komponenti na *Canvasu*, je prikazan na Slika 11.

Prilikom pokretanja scene UI kontrole ne bi trebale biti odmah vidljive, već je potrebo prikazivanje kontrola tek od trenutka stvaranja lika. Naime, stvaranjem lika unutar skripte za kontroliranje se na *Awake()* metodu instancira cijela UI komponenta, tj. unutar UI komponente dodana je skripta koja se ponaša kao *Singleton* te je omogućeno lijeno insanciranje.



Slika 11. Prikaz scene u *Unityu* sa UI elementima za kretanje lika.

4.5. Kreiranje sučelja za navigaciju između scena

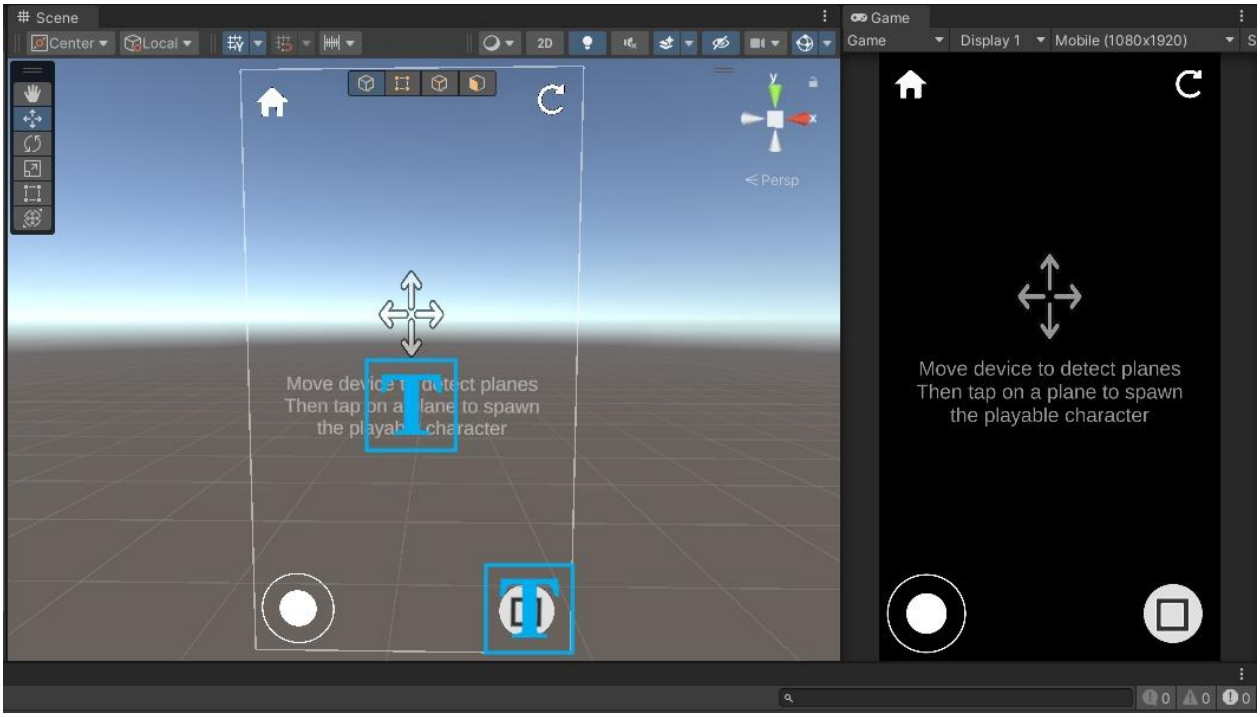
Kada su izrađene osnovne funkcionalnosti za igru, sljedeći korak je izraditi scenu glavnog izbornika te omogućiti navigaciju između scena. Za potrebe navigacije kreirana je jednostavna skripta koja koristi ugrađeni upravitelj scena za dinamičnu promjenu između scena i za izlaz iz aplikacije. Nakon izrađenog upravitelja scena dodaje se u glavnu scenu te se kreira nova scena koja će biti scena glavnog izbornika, tj. početna scena pri otvaranju aplikacije. Kada je kreirana nova prazna scena potrebno je unutar nje dodati *Canvas* objekt te po želji prilagođavati. U ovom slučaju na početnoj sceni se nalaze tri gumba, jedan za prebacivanje na sljedeću scenu, tj. glavnu scenu, jedan za prebacivanje na scenu sa ljestvicom rezultata, i jedan za izlaz iz aplikacije. U ovoj sceni se nalazi i naslov igre i pozadina po želji. Korišteni su fontovi preuzeti sa Dafont.com stranice, prilagođeni za *Unity* u *TextMeshPro Font Asset Creatoru*. Izgled početne scene se može vidjeti na Slika 12.

Slika 12. Prikaz početne scene u aplikaciji.

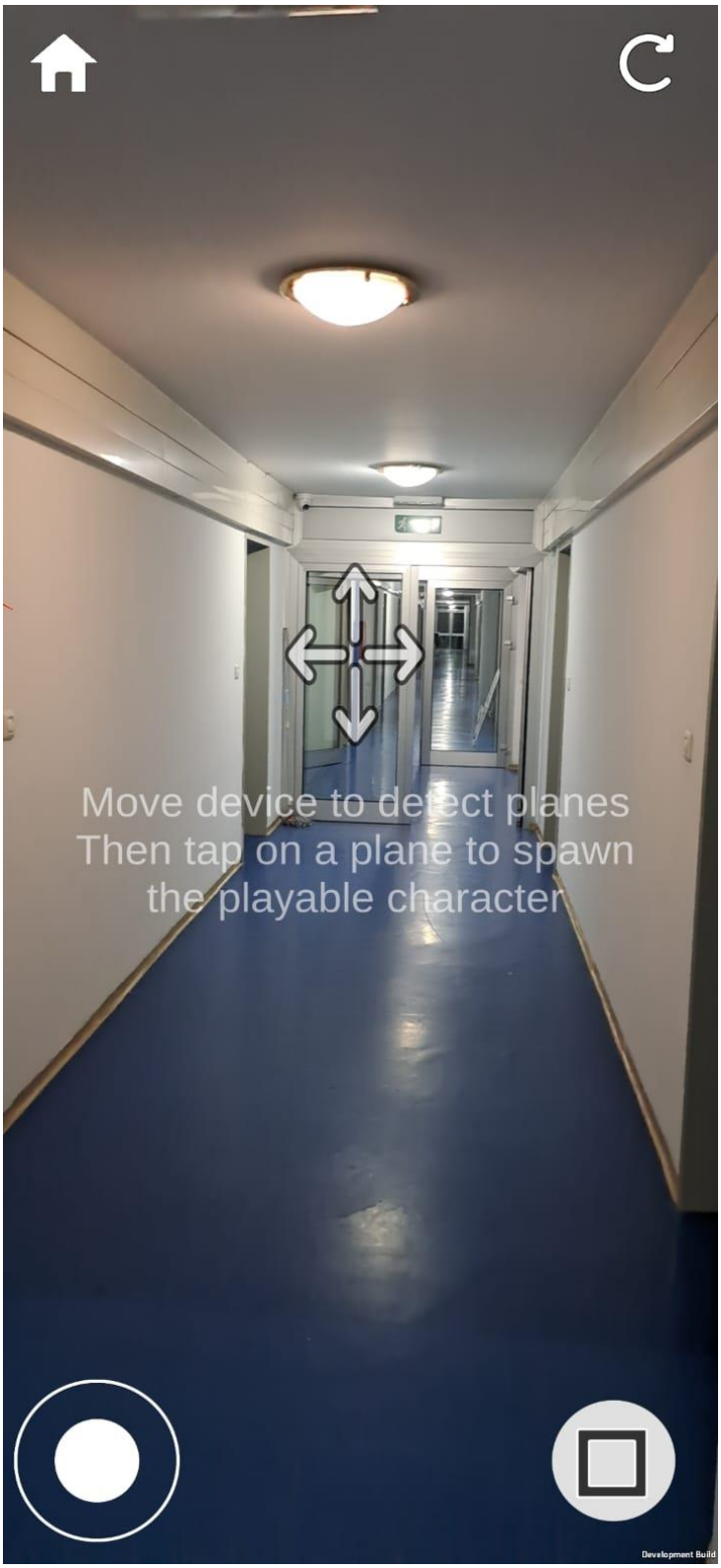
Navigacija između scena i izlaz iz aplikacije je omogućen kreiranjem upravitelja scena i u ovoj sceni te dodavanjem odgovarajućih metoda na *OnClick() listener* gumbova u sceni.

U glavnoj sceni su također dodani gumbovi za navigaciju u gornjim kutovima ekrana te su dodane upute kako koristiti aplikaciju prvim ulaskom na tu scenu. U lijevom gornjem kutu se nalazi simbol

kuće što predstavlja prelazak na početnu scenu, dok se u desnom gornjem kutu nalazi kružna strelica koja nakon pritiska na nju ponovo učitava glavnu scenu, tj. služi za „osvježavanje“ scene. Upute za korištenje se nalaze na sredini ekrana te nestaju nakon jednog pritiska na ekran. Cijeli dio uputa je gumb koji ima transparentnu pozadinu i razvučen je preko cijelog ekrana. Svi gumbovi su usidreni na određenoj strani u svrhu responzivnosti na ekranima različitih dimenzija. Na Slika 13 se može vidjeti prikaz glavne scene unutar *Unity* uređivača, a naSlika 14 ta ista scena je prikazana u proširenoj stvarnosti, tj. u mobilnog aplikaciji.

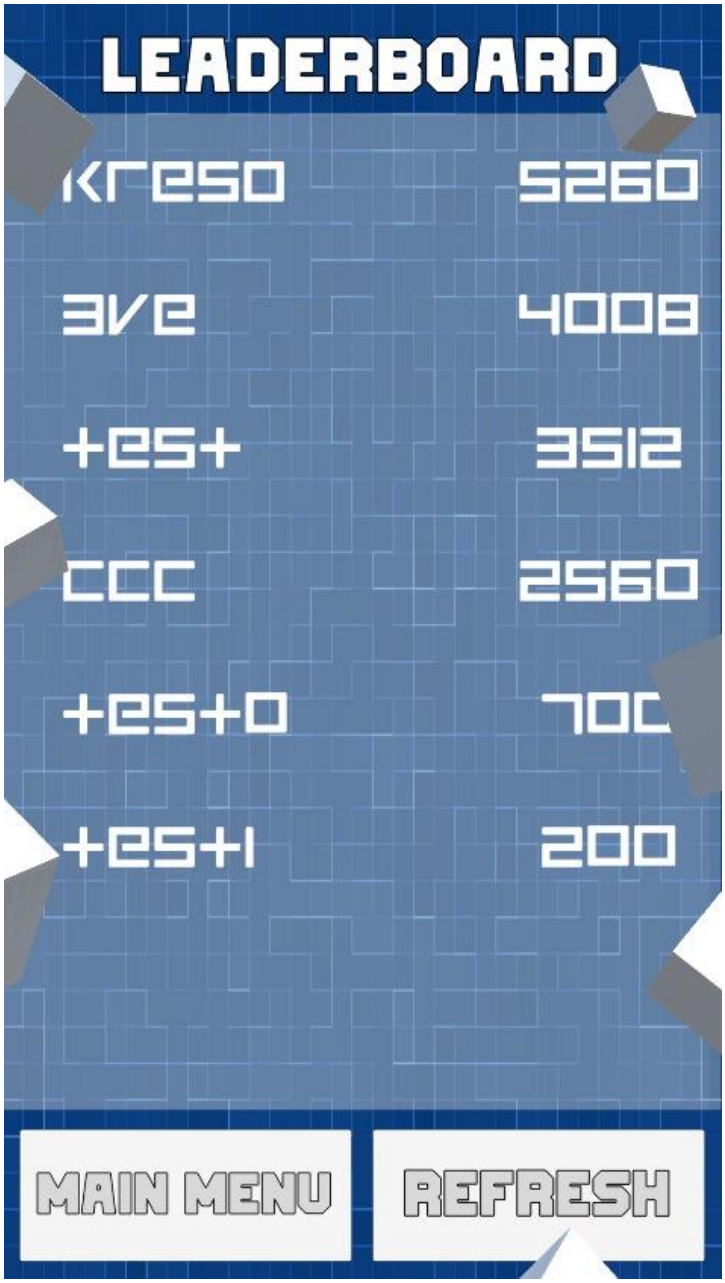


Slika 13. Prikaz glavne scene unutar *Unitya.*



Slika 14. Prikaz glavne scene na mobilnoj aplikaciji.

Scena sa ljestvicom rezultata je treća i konačna scena u igri. Postupak za kreiranje te scene je sličan kao za kreiranje scene glavnog menija. Kreira se prazna scena te unutar nje se doda *Canvas* element. Unutar tog elementa je dodan naslov na vrhu ekrana ispod kojeg se nalazi popis sa mogućnošću vertikalnog pomicanja koji služi kao ljestvica rezultata. Na dnu ekrana se nalaze dva gumba, jedan za povratak na glavni meni i drugi korišten za osvježavanja ljestvice rezultata (Slika 15).

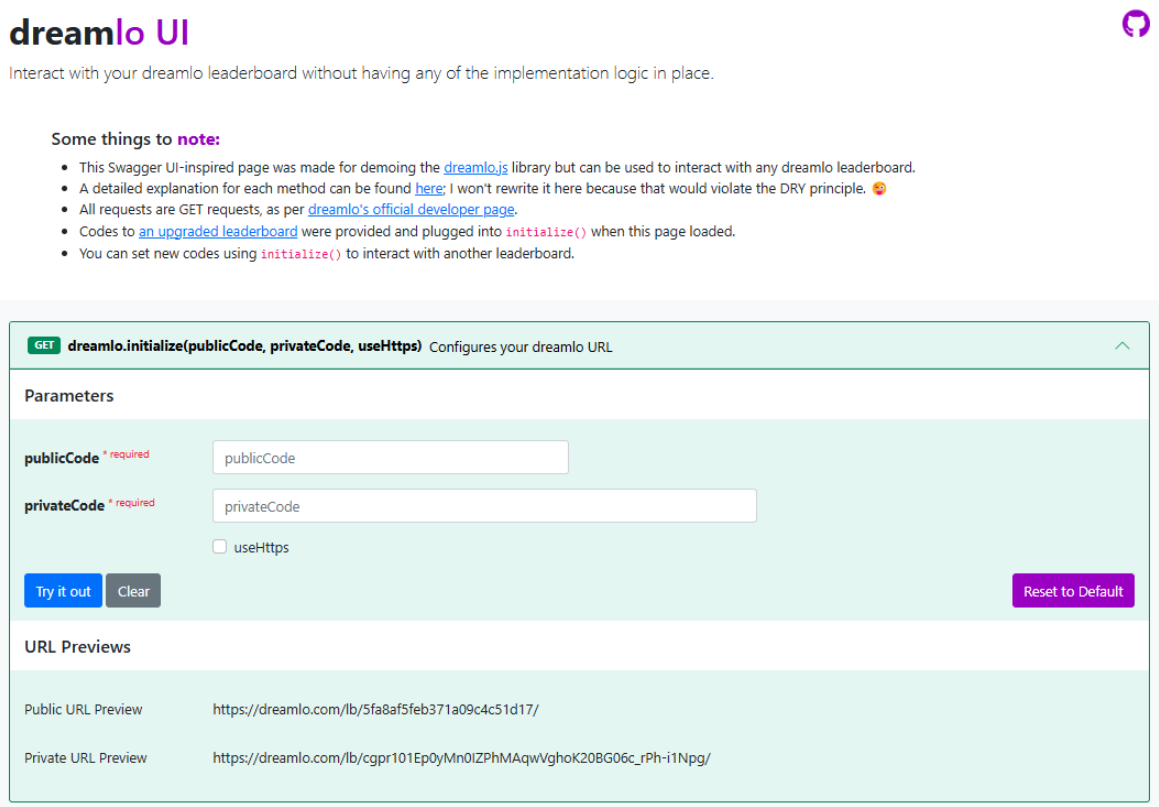


Slika 15. Prikaz scene sa ljestvicom rezultata.

4.6. Cilj igre

Kao cilj igre je skupiti sve tokene koje se automatski generiraju nakon stvaranja lika na detektiranoj površini u određenom vremenskom periodu. Tokeni se generiraju na temelju detektiranih površina, te se počinju generirati kada korisnik pritisne tipku na vrhu ekrana koja služi za zaustavljanje detekcije površina. Prvi korak izrade načina generiranja tokena je stvaranjem praznog objekta unutar kojeg se dodaje skripta pod nazivom *GameManager.cs*. Unutar ove skripte je način generiranja tokena, tj. instanciranje token *prefaba,* na nasumičnim koordinatama x i z osi, dok za y os se uzima vrijednost na kojoj se nalazi detektirana površina. Skripta također sadrži korutinu za odbrojavanje sekundi gdje se vrijednost u sekundama nadodaje na konačni rezultat nakon skupljenih svih tokena. Također je moguće odrediti količinu tokena koja će se stvarati te se može odrediti i udaljenost između tokena. U ovoj aplikaciji korišteno je vrijeme od 60 sekundi, a broj stvorenih tokena je deset. U slučaju da vrijeme istekne rezultat iznosi onoliko koliko je prikupljeno tokena, a ako se pokupe svi tokeni prije isteka vremena ostvaruje se rezultat ovisno o ostatku vremena. Kada se prikupe svi tokeni ili istekne vrijeme, pojavljuje se skočni prozor na ekranu unutar kojeg se nalazi konačni rezultat i polje za upisivanje imena da se spremi u bazu podataka. Skočni prozor je dio UI-a koji se onemogući učitavanjem scene te se nakon završetka igre ponovo omogući, a ako korisnik ne želi spremati svoj rezultat može kliknuti na osvježavanje scene te pokušati ponovo.

Baza podataka korištena u svrhu pohranjivanja rezultata ove aplikacije je *dreamlo. Dreamlo* je besplatni sustav za izradu ljestvica najboljih rezultata i promo kodova. Koristi jednostavne *HTTP GET (WWW)* zahtjeve, nije potrebno korištenje *PHP-a* ni *SQL-a.* Sustav može spremati ime, rezultat, vrijeme u sekundama i dodatnu znakovnu vrijednost, ali u svrhu ove aplikacije potrebno je spremanje samo imena i rezultata. Moguće ga je dodati putem *Unity Asset Storea* te upravljati njime putem web stranice čije sučelje je vidljivo na Slika 16 [13].



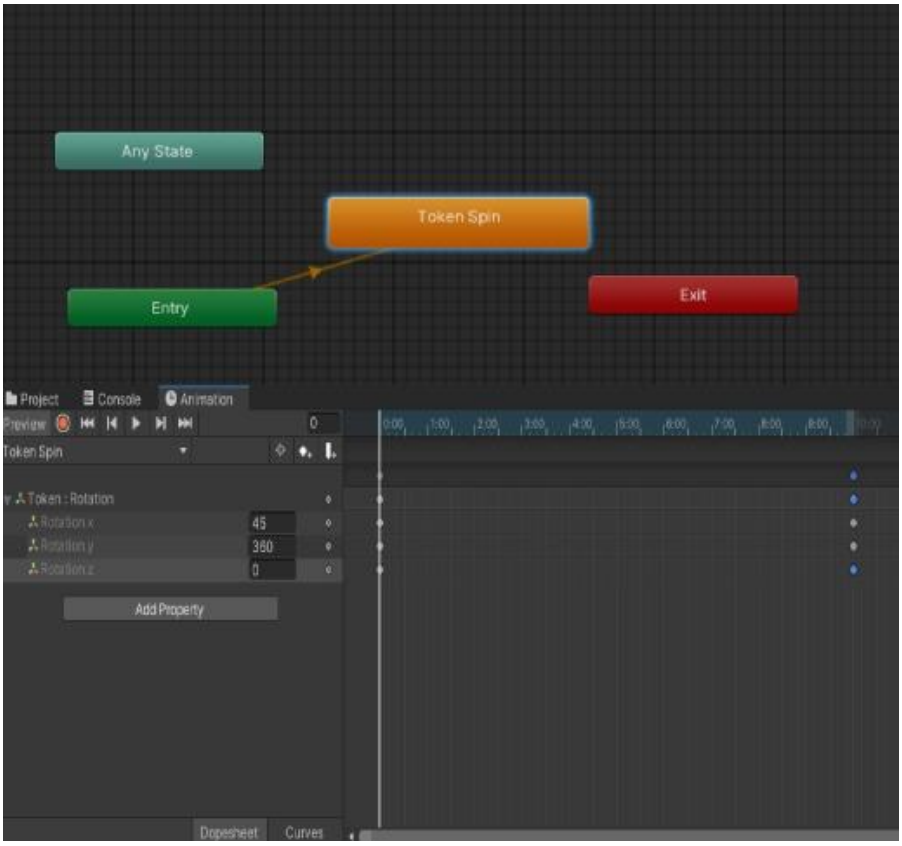
Slika 16. Prikaz *dreamlo web* sučelja [14].

4.7. Zvučni efekti

U svrhu uključivanja zvučnih efekata i pozadinske glazbe u *Unityu,* poboljšava se iskustvo igre te postaje privlačnije za igrače. Zvučni efekti dodani unutar aplikacije su tihi zvuk za akciju hodanja i za akciju skakanja. Također je dodan zvuk za prikupljanje tokena te upozorenje za skori istek vremena da igrač bude svjestan gubitka bodova. Kroz cijelu aplikaciju postoji i pozadinska glazba koju je moguće, zajedno sa svim zvučnim efektima, ugasiti u sceni glavnog izbornika. Zvučni izvor za akcije lika su dodane na objekt lika te unutar *CharacterMovement.cs* skripte su napisani određeni okidači, npr. u *Jump()* metodi se nalazi okidač za zvuk skakanja. Zvučni efekt za prikupljanje tokena je dodan na tokene te se okidač nalazi unutar token skripte. Pozadinska glazba se nalazi unutar UI objekta u sceni kako bi glazba uvijek bila prisutna neovisno o poziciji na kojoj se lik nalazi. Svi zvučni efekti se nalaze unutar besplatnog *Casual Game SFX* paketa koji je dostupan na *Unity Asset Storeu* [15].

4.8. Animacije

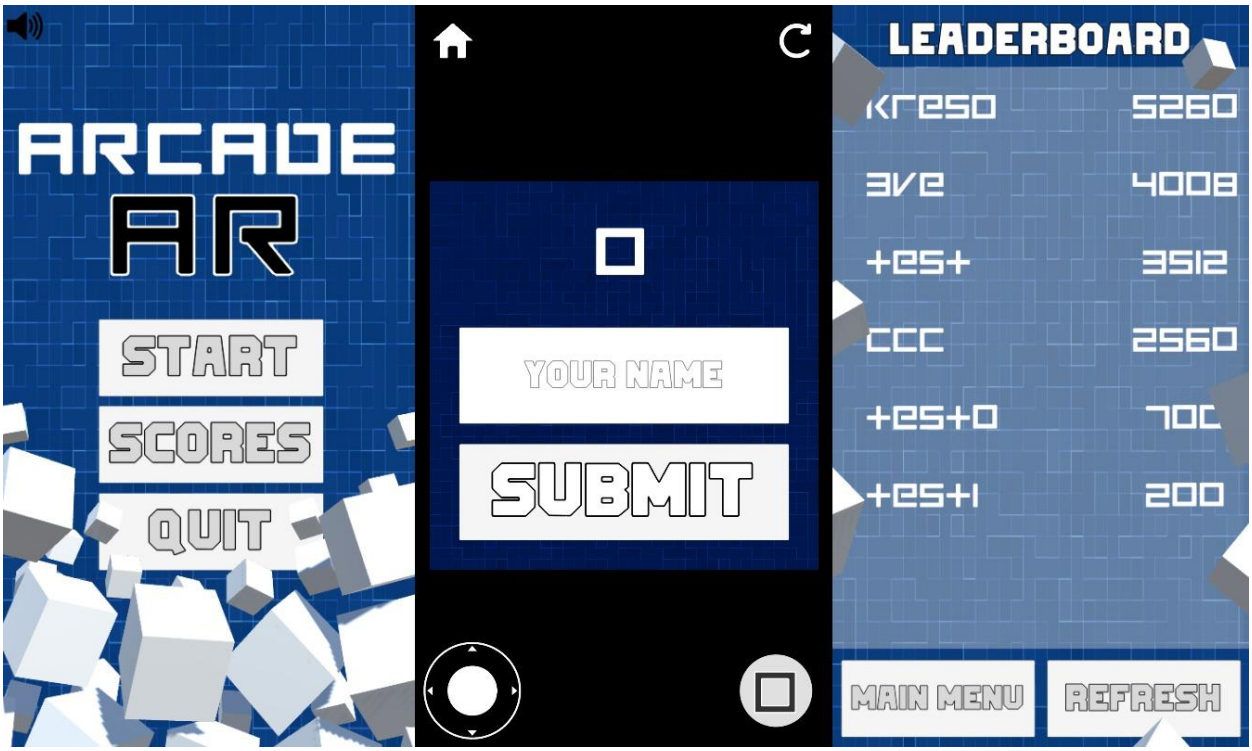
Kako bi lik i okolina bili dinamičniji i življi moguće je dodati animacije za poboljšanje iskustva igranja. Animacije u *Unityu* je moguće izraditi pomoću *Animator* komponente. Vrlo jednostavno je moguće izraditi složene animacije bez predubokog prethodnog znanja i iskustva. Moguće je animirati bilo koju komponentu, dio komponente ili više komponenti odjednom. U ovom radu *animator* će se koristit za jednostavne animacije kretanja lika i animacija tokena kako bi scena bila življa. Lik će se animirati na način da ako se kreće kroz scenu, kotač modela će se okretati pravovremeno i u slučaju skoka kotač će se pomjeriti vertikalno prilikom slijetanja kao imitacija amortizera. Animacija tokena će se sastojati od jednostavnog kružnog okretanja tokena oko vlastite osi za laganim vertikalnim pomjeranjem. Izgled uređivača animacije tokena je prikazana na Slika 17.



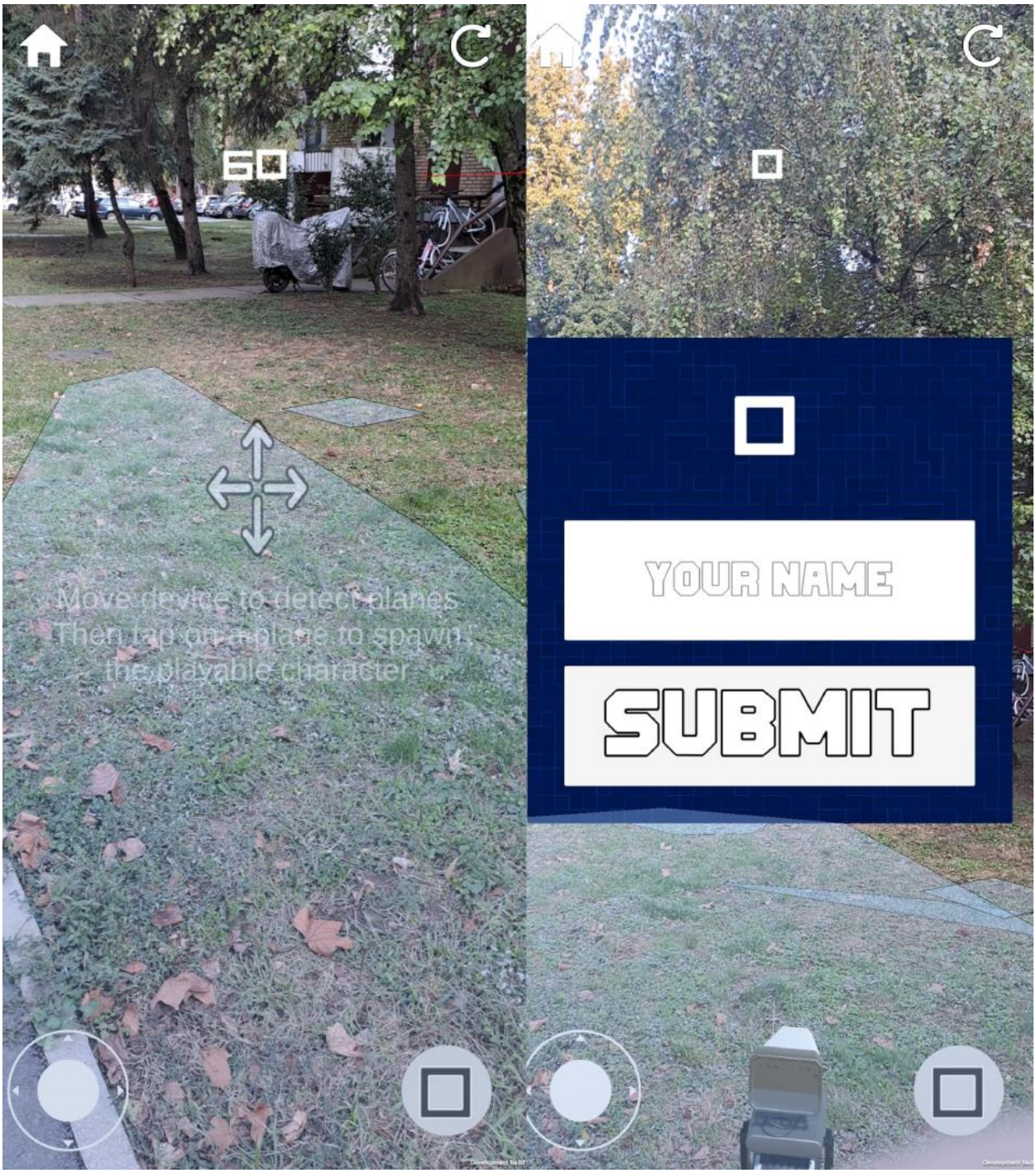
Slika 17. Prikaz uređivača animacije za token.

5. ANALIZA KONAČNOG REZULTATA

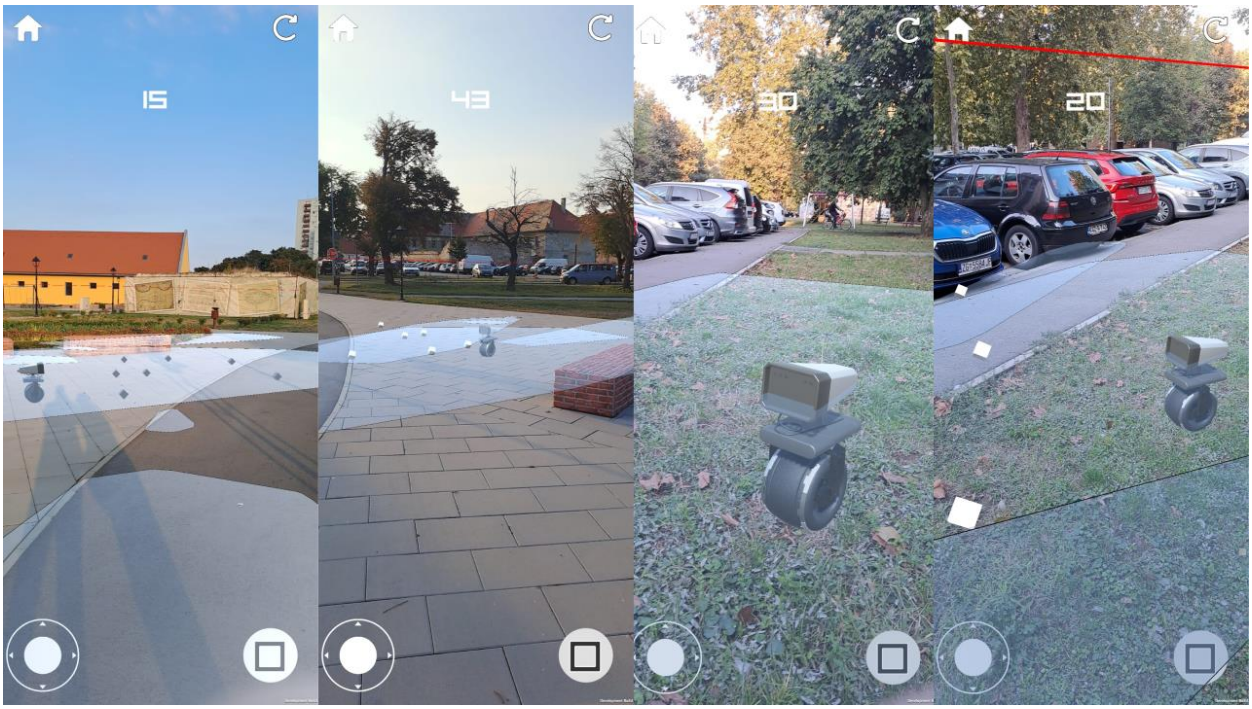
Konačni rezultat je igra koja se sastoji od tri scene: glavni meni, scena sa igrom i scena sa ljestvicom rezultata. Korisnikova prva interakcija je sa glavnom scenom unutar koje može navigirati na scenu sa igrom, scenu sa ljestvicom rezultata, izaći iz aplikacije i podesiti zvuk kroz cijelu aplikaciju. Prelaskom na scenu sa igrom, upali se zadnja kamera te se pojave upute na ekranu za pomjeranje mobilnog uređaja u svrhu detekcije ravnih površina u proširenoj stvarnosti. Kada je korisnik zadovoljan sa razinom koju je „kreirao“, tj. kada je zadovoljan sa površinama koje su detektirane, klikom na površinu može stvoriti svog lika kojeg će moći upravljati kroz scenu. Kada se stvori lik nije moguće više detektirati nove površine te ako korisnik nije zadovoljan sa razinom koju je kreirao, može kliknuti na gumb za resetiranje scene koji se nalazi u gornjem desnom kutu ekrana. Također kad se stvori lik, generiraju se tokeni kroz cijelu razinu te počinje odbrojavanje od 60 sekundi da se pokupe svi tokeni. Rezultat se računa na način da ako istekne vrijeme prije nego što su svi tokeni pokupljeni, broj pokupljenih tokena se množi sa sto. Ako su pokupljeni svi tokeni prije isteka vremena, također se pokupljeni tokeni množe sa sto, no na rezultat se dodaje preostalo vrijeme u sekundama pomnožene sa dvjesto. Cilj je da korisnik pokupi sve tokene u što bržem vremenu kako bi ostvario što bolji rezultat. Nakon što korisniku ili istekne vrijeme ili pokupi sve tokene sa viška vremena, pojavi se skočni prozor sa sučeljem za upisivanje imena i gumbom za pohranu imena i rezultata u bazu podataka, prikazan na Slika 18 zajedno sa svim ostalim scenama. Nakon unosa imena scena se resetira da korisnik, ako želi može ponovo igrati ili navigirati do scene sa ljestvicom rezultata da provjeri svoj položaj. Konačni izgled aplikacije je moguće vidjeti na slikama Slika 19 iSlika 20 kroz koje je prikazano detekcija površina te nakon detekcije, stvaranje lika i generirani tokeni za prikupljanje, te skočni prozor nakon što se pokupe svi tokeni ili istekne vrijeme.



Slika 18. Prikaz sve tri scene aplikacije.



Slika 19. Detekcija površina u stvarnom vremenu i skočni prozor.



Slika 20. Prikaz aplikacije u stvarnom vremenu na različitim udaljenostima.

6. ZAKLJUČAK

Cilj diplomskog rada je bila izrada prilagodljive arkadne igre u proširenoj stvarnosti gdje je korisniku omogućeno skeniranjem površina stvoriti lika te upravljati istim sa određenim ciljem savladavanja igre. Razina igre je predstavljena sa okruženjem korisnika te navigacijom lika kroz okruženja je potrebno skupiti sve tokene u najkraćem mogućem vremenu u svrhu ostvarivanja što bolje rezultata. Tehnologije korištene za izradu aplikacije su *Unity Game Engine*, te potrebni alati unutar *Unitya* za rad u proširenoj stvarnosti. Alati za tu svrhu su *XR Interaction Toolkit* čija je glavna svrha povezivanje svih interakcija u proširenoj stvarnosti, te *AR Foundation* koji omogućava osnovne funkcionalnosti za rad u proširenoj stvarnosti poput skeniranje površina i stvaranje *prefab* elemenata na tim površinama. Nakon izrade cijele aplikacije, pomoću jednostavnog *Unity* sučelja moguća je implementacija aplikacije na mobilni *Android* uređaj.

Za izradu aplikacije je potreban određen redoslijed koraka u svrhu što lakšeg testiranja i implementiranja. Unutar jedne scene su izrađivanje sve funkcionalnosti vezane za proširenu stvarnost, npr. stvaranje lika u proširenoj stvarnosti, dok su unutar druge scene izrađivane i testirane sve osnovne funkcionalnosti vezane za razvoj u *Unity3Du*, poput načina kretanje lika te generiranje tokena. Uz scene za testiranja kreirane su tri scene koje se nalaze u konačnom rezultatu: scena glavnog menija, scena sa igrom u proširenoj stvarnosti i scena sa ljestvicom rezultata.

LITERATURA

1. Službena stranica *PokémonGO* aplikacije: Pokémon GO (pokemongolive.com) [10.06.2024]
2. Stranica za pregled informacija o *PokemonGO* aplikaciji: A first look at Pokémon GO - PocketMonsters.Net [17.09.2024]
3. Stranica za pregled informacija o *Harry Potter: Wizards Unite* aplikaciji: Harry Potter: Wizards Unite | Harry Potter Wiki | Fandom [10.06.2024]
4. Službena stranica *Jurassic World Alive* aplikacije: Jurassic World Alive – Available now! Download Jurassic World Alive on iOS & Android [10.06.2024]
5. Službena stranica *SurgicalAR* aplikacije: A new standard of surgery using augmented reality. (medivis.com) [10.06.2024]
6. Službena stranica *MergeEDU* aplikacije: Hands-on Learning with Interactive 3D Models & Simulations (mergeedu.com) [10.06.2024]
7. *X* profil *Medivis* studia: MEDIVIS (@Medivis_AR) / X [17.09.2024]
8. Stranica dokumentacije *Unity Game Enginea*: Unity Documentation [13.06.2024]
9. Stranica dokumentacije *XR Interaction Toolkita*: XR Interaction Toolkit | XR Interaction Toolkit | 3.0.4 (unity3d.com) [13.06.2024]
10. Stranica dokumentacije *AR Foundationa*: AR Foundation | AR Foundation | 6.0.2 (unity3d.com) [13.06.2024]
11. Stranica dokumentacije stanja praćenja poze: Enum TrackingState | Package Manager UI website (unity3d.com) [14.06.2024]
12. Stranica dokumentacije *AR Placement Interactablea*: AR Placement Interactable | XR Interaction Toolkit | 2.0.4 (unity3d.com) [15.06.2024]
13. *Unity Asset Store* stranica *dreamlo* sustava ljestvica: dreamlo.com - Free Instant Leaderboards and Promocode System | Network | Unity Asset Store [16.06.2024]
14. Službena *dreamlo* stranica: dreamlo UI [17.09.2024]
15. *Unity Asset Store* stranica *Casual Game SFX* paketa: FREE Casual Game SFX Pack | Audio Sound FX | Unity Asset Store [16.06.2024]
16. *Sketchfab* stranica: Sketchfab - The best 3D viewer on the web [04.09.2024]

POPIS SLIKA

SAŽETAK

Rezultat ovog diplomskog rada je korištenjem tehnologije proširene stvarnosti izraditi igru koja spaja fizički i virtualni svijet. Bilo je potrebno izraditi mobilnu aplikaciju kroz koju je moguće imati interakciju virtualnog lika sa površina u stvarnome svijetu. Izrada aplikacije se sastojala od više dijelova, rad sa proširenom stvarnošću te omogućavanje svih interakcija lika sa okolinom. Korištenjem *XR Interaction Toolkita* i *AR Foundationa* omogućen je jednostavan rad sa proširenom stvarnošću unutar *Unitya,* dok se drugi aspekt rada temelji na poznavanju i vještinama sa *Unity3D* programiranjem te osnovnim principima objektno orijentirane paradigme. Krajnji rezultat je aplikacija unutar koje je moguće stvoriti lika na detektiranoj površini te upravljanje istim u svrhu skupljanja svih tokena koji se generiraju u proširenoj stvarnosti u što manjem vremenu da bi se postigao što bolji rezultat. Potencijalni problem unutar aplikacije jest ovisnost snage mobilnog uređaja o detekciji površina, gdje slabiji uređaji će znatno sporije detektirati površine te će to remetiti korisničko iskustvo.

Ključne riječi: AR, arkadna igra, detekcija površina, igra u proširenoj stvarnosti, mobilna aplikacija

ABSTRACT

The result of this thesis is to use augmented reality technology to create a game that combines the physical and virtual worlds. It was necessary to create a mobile application through which it is possible to have the interaction of a virtual character with planes in the real world. The development of the application consisted of several parts, working with augmented reality and enabling all interactions of the character with the environment. Using the XR Interaction Toolkit and AR Foundation enables simple work with augmented reality within Unity, while the other aspect of work is based on knowledge and skills with Unity3D programming and the basic principles of the object-oriented paradigm. The result is an application within which it is possible to create a character on the detected surface and manage it to collect all the tokens that are generated in augmented reality in the shortest possible time to achieve the best possible result. A potential problem within the application is the dependence of the power of the mobile device on the plane detection, where weaker devices will detect planes much slower, which in end will disrupt the user experience.

Keywords: AR, arcade game, augmented reality game, mobile application, plane detection

ŽIVOTOPIS

Krešimir Matić rođen je 14.06.2000. godine u Zagrebu, s prebivalištem u općini Vitez, Bosna i Hercegovina. Osnovnu i srednju školu je završio u Vitezu, te 2019. godine se upisuje na Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku na preddiplomski sveučilišni studij Računarstvo. 2022. godine upisuje se na sveučilišni diplomski studij Računarstvo, izborni blok Informacijske i podatkovne znanosti.

Tijekom osnovnoškolskog i srednjoškolskog obrazovanja sudjelovao je na brojnim natjecanjima iz matematike, te je u 4. razredu srednje škole osvojio drugo mjesto na školskom natjecanju u Mostaru. Učio je i usavršio sporazumijevanje na engleskom jeziku tijekom cijelog školovanja.

_____

Potpis autora