

RAZISKOVALNA NALOGA

**APLIKACIJA V NAVIDEZNI
RESNIČNOSTI KOT PRODAJNO
ORODJE**

Avtor:

Marko GORENAK, M-4. e

Mentor:

Alen PAVŠAR, ing. multimedije

Mestna občina Celje, Mladi za celje

Celje, 2017

POVZETEK

Za raziskovalno nalogo smo si zastavili cilj, da ustvarimo aplikacijo, ki bo podjetju MIK Celje, d.o.o., pripomogla k oglaševanju in prodaji oken v navidezni resničnosti. Interaktivna aplikacija je bila izdelana med praktičnim izobraževanjem v podjetju Medijska produkcija Almedia.

Pri ustvarjanju smo si zastavili naslednje cilje:

- narediti inovativen izdelek z uporabo povsem nove tehnologije,
- aplikacijo prvič predstaviti na Mednarodnem obrtnem sejmu v Celju leta 2016,
- z izdelavo pridobiti novo znanje na področju programiranja v programskem okolju Unity.

Vedli smo, da ne bo enostavno, a smo želeli pridobiti izkušnje iz realnega okolja. Način izvedbe takšnega projekta je povsem drugačen od šolskega. Okolje je dinamično, časa je bilo izredno malo, ker je naročnik želel izdelano aplikacijo do Mednarodnega obrtnega sejma. Prvi korak k izdelavi je bila analiza obstoječe tehnologije in dejstev, kaj lahko v tako kratkem roku naredimo. Za izdelavo aplikacije smo imeli tri mesece časa. Ko smo določili strojno in programsko opremo, ki jo bomo med izdelovanjem uporabljali, smo z ekipo sestavili zahteven načrt in ga po najboljših močeh tudi uresničili.

KAZALO

1 UVOD.....	1
1.1 HIPOTEZE.....	1
2 STROJNA OPREMA.....	2
2.1 MIŠKA.....	2
2.2 TIPKOVNICA.....	3
2.3 ZASLON.....	3
2.4 KONZOLA TRUST PS3.....	4
2.5 LEAP MOTION.....	5
2.6 OCULUS RIFT.....	6
3 PROGRAMSKA OPREMA.....	7
3.1 UNITY.....	7
3.2 MONODEVELOP.....	8
4 INTERAKTIVNA APLIKACIJA V NAVIDEZNI RESNIČNOSTI	9
4.1 IDEJA IN INTERAKTIVNOST V APLIKACIJI.....	9
4.2 NAVIDEZNA RESNIČNOST.....	11
4.3 VZOREC HIŠE IN IZDELAVA 3D MODELA.....	12
4.3.1 Oblikovanje 3D Objektov v Cinema 4D.....	12
4.3.2 Integracija 3D objektov v Unity.....	12
4.4 NAČRTOVANJE UPORABNIŠKE IZKUŠNJE.....	13
4.4.1 Načrt Programske Arhitekture Aplikacije.....	13
4.4.2 Oblikovanje Uporabniškega vmestnika in grafičnih elementov.....	13
4.4.3 Postavljanje Okolja.....	14
4.4.4 Pisanje Kode.....	14
4.4.4.1 Glavni Mani.....	14
4.4.4.2 Kamera Uporabnika.....	15
4.4.4.3 Window Manager.....	15
4.4.4.4 Application Controller.....	16
4.4.5 Izdelava Interaktivnega okna.....	16

4.4.6 Menjava oken, barva in materialov.....	17
4.4.7 Menjavanje Senčil.....	17
4.4.8 Programiraje Senzorja Leap Motion.....	18
4.4.9 Programiranje Konzole.....	18
4.5 IZDELAVA NAVODIL ZA UPORABO.....	19
4.6 TESTIRANJE.....	20
4.7 PUBLISH.....	20
5 POTRDITEV HIPOTEZ.....	21
6 ZAKLJUČEK.....	22

KAZALO SLIK

SLIKA 1: Miška.....	2
SLIKA 2: Tipkovnica.....	3
SLIKA 3: Zaslon.....	3
SLIKA 4: Krmilnik.....	4
SLIKA 5: Leap Motion.....	5
SLIKA 6: Oculus Rift Očala.....	6
SLIKA 7: Unity 5 Logo.....	7
SLIKA 8: Mono Develop.....	8
SLIKA 9: Interaktivnost v Aplikaciji.....	9
SLIKA 10: Leap Motion v Uporabi.....	10
SLIKA 11: Navidezna Resničnost.....	11
SLIKA 12: Načrt glavnega menija za aplikacijo.....	12
SLIKA 13: GlavniMeni.....	13
SLIKA 14: KameraUporabnika.....	14
SLIKA 15: WindowManager.....	14
SLIKA 16: ApplicationController.....	15
SLIKA 17: Enokrilno Okno.....	16
SLIKA 18: Meni Materialov.....	17
SLIKA 19: Gumb za Senčila.....	17
SLIKA 20: Vnos.....	18
SLIKA 21: Navodila.....	19
SLIKA 22: Izvoz.....	20

1 UVOD

Za aplikacijo v navidezni resničnosti smo se odločili, ker smo želeli pridobiti več znanja in raziskati, kako bi ustvariti aplikacijo z različnimi interaktivnimi orodji, saj se trg nenehno razvija in je treba slediti novi tehnologiji. Sodobno podjetje mora kot trženjsko orodje uporabljati sodobno tehnologijo. Ne le orodja na spletu, temveč tudi aplikacije.

Največja vprašanja so bila:

Kako se bomo približali občutku, ki ga doživljamo v realnosti?

Bo uporabnik verjel doživetju v VR?

Kako lahko kasneje aplikacijo nadgradimo?

Pozicioniranje aplikacije v svetu (regionalno in globalno).

1.1 HIPOTEZE

V okviru raziskovalne naloge smo si zastavili sledeče hipoteze:

1. Interaktivna aplikacija služi kot prodajno orodje za komercialista.
2. V navidezni resničnosti se lahko uporabnik premika kot v realnosti.
3. Aplikacija omogoča interakcijo z virtualnimi objekti.
4. Objekti v virtualni resničnosti so zelo realistični in podobni tistim v realnosti.

2 STROJNA OPREMA

Za izdelovanje aplikacije potrebujemo ustrezno strojno opremo, saj bosta od nje odvisna tako razvoj kot uporaba aplikacije.

2.1 MIŠKA

Miška (vhodna naprava) je eden glavnih delov strojne opreme, če želimo urejati v programih, ki so opotrebni za izdelavo aplikacije. Za njeno delovanje ne potrebujemo dodatne programske opreme.



Slika 1: Miška

2.2 TIPKOVNICA

Je eden glavnih delov strojne opreme, saj z njo vnašamo simbole v najrazličnejša okna. Za njeno delovanje ne potrebujemo dodatne programske opreme.



Slika 2: Tipkovnica

2.3 ZASLON

Zaslon je, kot miška in tipkovnica, enako pomemben del strojne opreme. Zaslon nam vizualno sporoča, da premikamo po programih.



Slika 3: Zaslon

2.4 KONZOLA TRUST PS3

Konzola je uporabljena v aplikaciji kot interaktivno orodje. Z njo sporočamo aplikaciji, kam se želimo premakniti v navidezni resničnosti, da premikamo skozi svet. Z njim odpiramo menija, vklopimo tutorial in glasbo. Konzola se priključi z navadnim usb priključkom v usb vhod na računalniku, za uporabo v aplikaciji moramo skozi Unity, narediti povezave tipk, da jih lahko uporabimo v kodi za sprožitev različnih ukazov.



Slika 4: Krmilnik

2.5 LEAP MOTION

Je senzor, ki zazna roke in informacije poda računalniku, uporabimo ga kot krmilnik. Če ga želimo uporabljati v tem programskem okolju, si Leap Motion SDK in posebne datoteke, za delovanje v programu Unity, prenesemo z njihove spletne strani. Naprava se v tem primeru namesti na očala, priključena pa je v računalnik z navadnim usb-jem. To napravo smo si izbrali, da uporabnika čim bolj približamo v svet aplikacije, k temu veliko pripomore videnje in uporabljanje svojih rok, povezava z aplikacijo je zato veliko enostavnejše. Ugotoviti smo morali, na kakšen način se bo uporabnik povezal z našo aplikacijo skozi Leap Motion, da ne bo težav. Zato smo se odločili, da se uporabnik poveže z objekti skozi meni, ki ga nadzira s svojimi rokami. Tako se izognemo problemu dotikanja predmeta, saj v resničnem svetu tega predmeta pred uporabnikom ni. Tudi če bi se hotel povezati z objekti z rokami, bi prišlo do težav, saj virtualne roke nosijo neskončne sile, kar bi pripeljalo do tega, da se nekateri objekti ne bi vedli realistično.



Slika 5: Leap Motion

2.6 OČALA OCULUS RIFT

Oculus rift očala so naprava za izkušanje navidezne resničnosti. Oculus Rift so eno izmed podjetij, ki izdeluje opremo za navidezno resničnost. Za uporabo imajo senzor, ki zazna, kako visoko in kje je kamera (v malem prostoru), očala pa zaznajo rotacijo, imajo pa tudi slušalke. Očala so priključena na računalnik s HDMI in usb kablom, senzor pa samo z usb, povezan pa je tudi neposredno z očali. Senzor po navadi nastavimo nekje v višino glave, najpogosteje kar na računalniški zaslon. Očala lahko prilagajamo za različne velikosti glav uporabnikov.



Slika 6: Oculus Rift Očala

3 PROGRAMSKA OPREMA

3.1 PROGRAMSKO OKOLJE UNITY

Unity je program, ki se uporablja za izdelovanje video iger, simulacij in aplikacij. Je eden izmed game enginev in je v krogu z najboljšimi. Delali smo z brezplačno verzijo (Unity 5.3.4 p1), saj je imela vse, kar smo potrebovali za aplikacijo. Unity smo izbrali zato, ker je enostavna za uporabo in hitro pridobiš prototip, da lahko vidimo, kaj bo šlo in kaj ne in popravimo napake, videz, spremenimo objekte, skripte.

Za uporabo potrebujemo vsaj:

- Windows 7 +, Mac OS X 8.10 +
- Grafično kartico z DX9 ali DX11

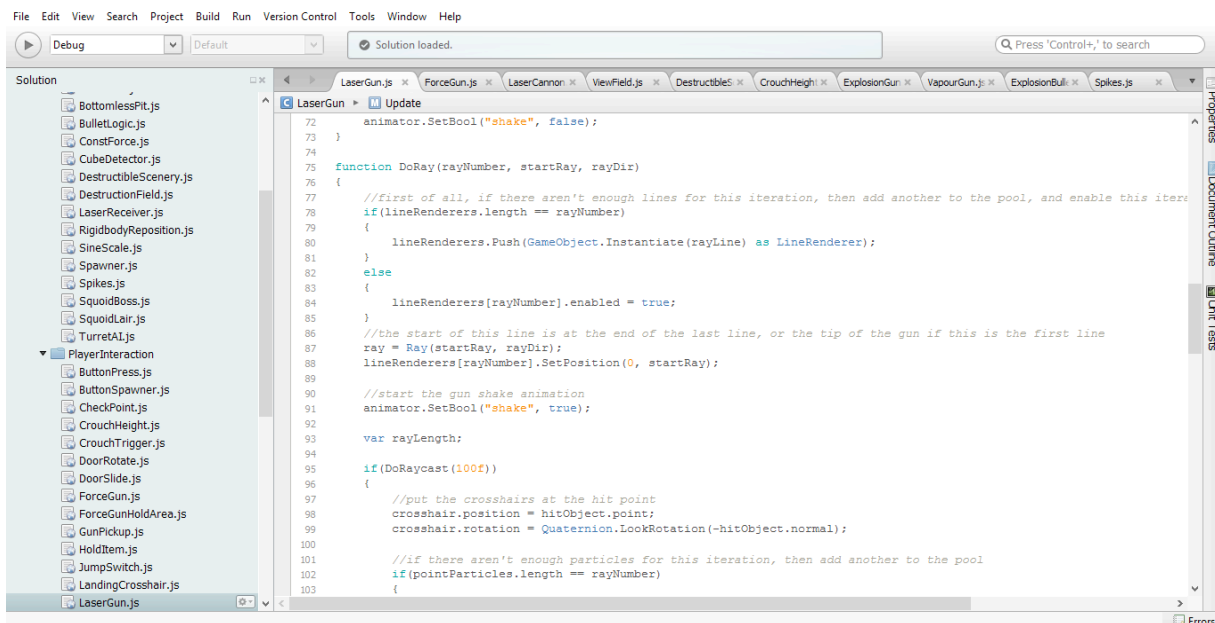
Ostalo pa je odvisno od velikosti projekta.



Slika 7: Unity 5 Logo

3.2 MONODEVELOP

Se uporablja kot programsko orodje za pisanje kode programskih jezikov. Monodevelop je del programskega okolja Unity in je privzeta opcija, lahko pa za namestitev monodevelop izberemo mictosoft visual studio. Izbrali smo monodevelop, ker ga že dlje uporabljamo in je bolj prilagojen.



Slika 8: Mono Develop

4 INTERAKTIVNA APLIKACIJA V NAVIDEZNI RESNIČNOSTI

4.1 IDEJA IN INTERAKTIVNOST V APLIKACIJAH

Glavni cilj pri izvedbi je bil, da ima uporabnik v aplikaciji kontrolo nad premikanjem pozicije, premikanjem kamere in interakcijo virtualnega sveta s svojimi rokami. Sprva smo hoteli, da bi lahko z gibom roke ukazali prestavitev naprej, saj je prostor gibanja z očali omejen, a ker zaznavanje gibov rok z Leap Motionom še ni najboljše, smo uporabili konzolo s katero se premikaš z premikanjem joysticka. Leap Motion uporabljamo za menije in za premikanje. V prihodnosti bi lahko dodali premikanje z gibi rok, da bi se znebili uporabe konzole in se s tem še bolj približali interaktivnosti in realističnosti.



Slika 9: Interaktivnost

Uporabniku smo želeli uporabo navidezne resničnosti prikazati, kako so videti MIK-ova okna vgrajena v hiši, da bi si lahko ustvaril bolj realistično vizijo končnega videza v svojem stanovanju.

Z interaktivnostjo v aplikaciji lahko uporabnik upravlja in se odloča, kaj želi videti in s tem spreminja vsebino. Načinov implementacije interaktivnosti je veliko, kot je velik tudi razpon, kako velik vpliv in kontrolo ima uporabnik nad interaktivno vsebino.



Slika 10: Leap Motion v uporabi

4.2 NAVIDEZNA RESNIČNOST

Navidezna resničnost omogoča uporabnika približati svetu, ki smo ga ustvarili. Vanj se lahko bolj uživi in počuti kot del virtualnega sveta. Objekt lahko opazuje iz vseh zornih kotov, ob čemer lahko dodajamo različne efekte, npr. lebdeče informacije, zamenjuje lahko materiale, iz katerih so okna, dodajanja in odvzema elemente, česar v realnem svetu ne bi mogli doseči. Tako prodajalec z lahkoto stranki pokaže in razloži svoj produkt.



Slika 11: Navidezna resničnost

4.3 VZOREC HIŠE IN IZDELAVA 3D MODELA

Od naročnika smo dobili reference in načrte za virtualno hišo, v kateri je želel imeti razstavljen svoja okna. V 3D program Cinema 4D smo vnesli sliko in po njej začeli izdelovati aplikacijo.

4.3.1 Oblikovanje 3D objektov v programu cinema 4D

Za izdelavo 3D modelov smo uporabili program Cinema 4D. Pri izdelavi 3D modelov programu ni potrebno imeti dobrega upodabljanja, saj bo to potekalo v pravem času v sami aplikaciji. 3D program pa nam mora omogočati hitro delo in dobra orodja za lažjo in kakovostno izdelavo 3D modelov.

4.3.2 Integracija 3D objektov v Unity

Unity sprejme veliko 3D datotek, ki jih lahko takoj uporabimo v sceni. Datoteke se razlikujejo po vsebini, nekatere prenesejo samo 3D model, lahko pa tudi vsebujejo texture in animacije. Za izvažanje 3D modelov, je najbolj znan in najpogosteje uporabljen format .fbx in .obj, saj izvozi samo manjše datoteke s podatki, ki jih potrebujemo. Izvoz 3d datotek je lahko počasen postopek, če samo izdeluješ prototipe. Lahko pa v Unity neposredno uvozimo programske datoteke iz programa (npr. .max, .mb, .blend ...).

V našem projektu smo 3d objekte izvozili v format .fbx, saj smo imeli veliko 3D modelov, ki niso smeli zavzeti veliko prostora.

4.4.3 Postavljanje okolja

Načrt za okolje smo ustvarili v beležko, napisali komponente aplikacije in jih razvrstili po pomembnosti. Najprej smo pripravili Oculus DK2 (Developer Kit 2), tako da smo prenesli Oculus SDK (Software Development Kit), kot tudi nekatere datoteke, ki so potrebne za delovanje v programu Unity.

4.4.4 Pisanje kode C#

C# (see sharp) je programski jezik, ki se največkrat uporablja v Unity. C# je večparadigemski programski jezik, ki obsega močo tipizacijo ter imperativno, deklarativno, funkcijsko, generično, komponentno orientirano in objektno orientirano programiranje in vsebuje zmožnost refleksije. Upravljanje s pomnilnikom temelji na čistilniku spomina.

Kodo smo napisali v programskem jeziku C# v programu monodevelop. Napisali smo različne skripte, ki služijo npr. premikanju, odpiranju oken, uporabi menija ...

4.4.4.1 Glavni meni

MainManu je skript za glavni meni. Poskrbi za menije, spreminjanje materialov oken, spreminjanje barve fasade in hitro premikanje po sobah. Prejema in oddaja informacije k skriptu UserCamera in Application Controller (skripti navedeni spodaj).

```
//change material
case 2:
    allMaterial.CopyPropertiesFromMaterial (materials [caseNum]);
    break;
//open manu
```

Slika 13: Glavni meni

4.4.4.2 Kamera uporabnika (UserCamera)

Poskrbi za raycasting, ki zazna, kaj gledamo in te informacije odda različnim skriptom, da lahko odpiramo elemente in se povežemo z ostalimi elementi.

```
//Raycating function.
void Raycasting()
{
    RaycastHit hit;
    Vector3 forward = transform.TransformDirection (Vector3.forward) * 10;
    Debug.DrawRay(transform.position, forward, Color.green);

    //Raycast from Eyes for detecting objects.
    if (Physics.Raycast (transform.position, (forward), out hit, 10, toHit)) {
        distanceFrom = hit.distance;
        currObj = hit.transform.gameObject;
        Obje = hit.transform.gameObject;

        //for Windows
        if(hit.transform.tag == "Window")
        {
```

Slika 14: UserCamera

4.4.4.3 Upravljaliec oken (Window Manager)

Ta skript poskrbi za odpiranje oken. Ko KameraUporabnika sporoči, da uporabnik gleda v okno, ga lahko odpre z ui elementi. Podobni skipti so še za vrata in drsna vrata, ki se odpirajo drugače.

```
void Update () {
    //Update handle,window rotation.
    handle.transform.localEulerAngles = new Vector3 (handleRotX + userSliderHandle , handleRotY, handleRotZ) ;
    window.transform.localEulerAngles = new Vector3 (windowRotX, windowRotY + userSliderWindow, windowRotZ+ userSliderUpright) ;
```

Slika 15: WindowManager

4.4.4.4 Krmilnik aplikacije (Application Controller)

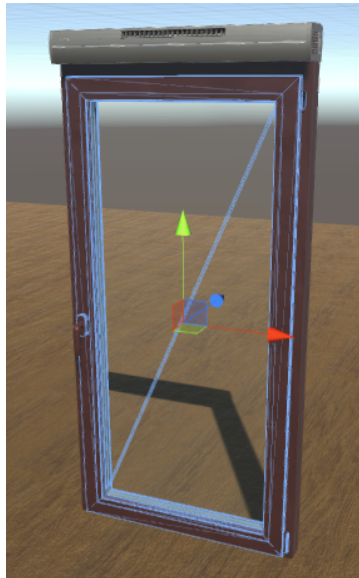
Poskrbi za input kontrolerja, da lahko uporabnik odpre različne menije s pritiskom gumba, prav tako vključi in izključi moteče elemente aplikacije, npr. glasba, informacije, meniji.

```
transform.rotation = Quaternion.Euler (transform.rotation.x , cameraRot , transform.rotation.z);
if (isRight) {
    RB.AddForce (transform.TransformDirection(Vector3.forward) * Input.GetAxis(buttons[0]) * speed);
    RB.AddForce (transform.TransformDirection(Vector3.right) * Input.GetAxis(buttons[1]) * speed);
} else {
    RB.AddForce (transform.TransformDirection(Vector3.forward) * Input.GetAxis(buttons[2]) * speed);
    RB.AddForce (transform.TransformDirection(Vector3.right) * Input.GetAxis(buttons[3]) * speed);
}
```

Slika 16: ApplicationController

4.4.5 Izdelava interaktivnega okna

Za okno smo napisali skript (WindowManager), ki poskrbi, da ga lahko v meniju ročno odpiramo z drsnikom. Meni se pokaže, ko UserCamera skript zazna okno. Objekt smo shranili v knjižnico že izdelanih modelov, da ga lahko vstavljamo, kamor želimo, in da obdrži vse prvotne nastavitve, ki jih lahko pozneje spremenimo.



Slika 17: Enokrilno okno

4.4.6 Menjava oken, barv in materialov

Za menjavo materialov smo naredili meni, v katerem z virtualno roko izberemo material s pritiskom na gumb. Za to je napisana koda v MainMenu, materiale lahko kadarkoli zamenjamo za druge.



Slika 18: Meni za materiale

4.4.7 Menjava senčil

S pritiskom na gumb lahko zamenjamo objekt (rolete ali žaluzije). Za to poskrbi skript MainMenu, ki objekt vključi ali izključi.



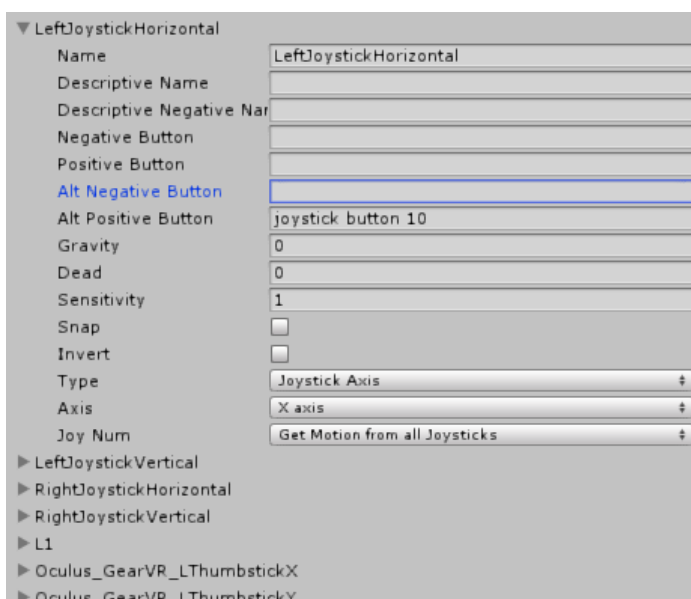
Slika 19: Gumb za senčila

4.4.8 Programiranje senzorja Leap motion

Nato smo začeli raziskovati novo napravo Leap Motion, ki jo bomo uporabili za adicionalno interaktivnost v aplikaciji. Pogledali smo, kako implementirati napravo v program Unity in si prenesli SDK, da bomo lahko leap napravo implementirali v kodo. Pomagali smo si s spletno stranjo Leap Motion, na kateri so navodila za začetek, in s forumi, kjer smo poizvedeli za rešitve, ko česa nismo razumeli ali ko se je kje zataknilo.

4.4.9 Programiranje konzole

Konzolo smo vključili v projekt tako, da smo pod Vnosom (Input) spremenili podatke, ki so se prilagodili gumbom konzole. S tem lahko hitro spremenimo kontrole.



Slika 20: Vnos

4.5 IZDELAVA NAVODIL ZA UPORABO

Pri izdelavi navodil smo vključili že navedena navodila za strojno opremo (Oculus Rift, Leap Motion) in dodali navodila za uporabo aplikacije (npr. uporaba gumbov na konzoli).

Oculus Rift opozarja, da ta oprema ni primerna za otroke in osebe, ki jih obide slabost pri premikanju, saj se lahko v aplikaciji premikaš, v realnosti pa se ne. Leap Motion pa vključuje navodilo, da moraš pri uporabi imeti roke v prostoru, ki ga senzor zazna (področje rok pred senzorjem).



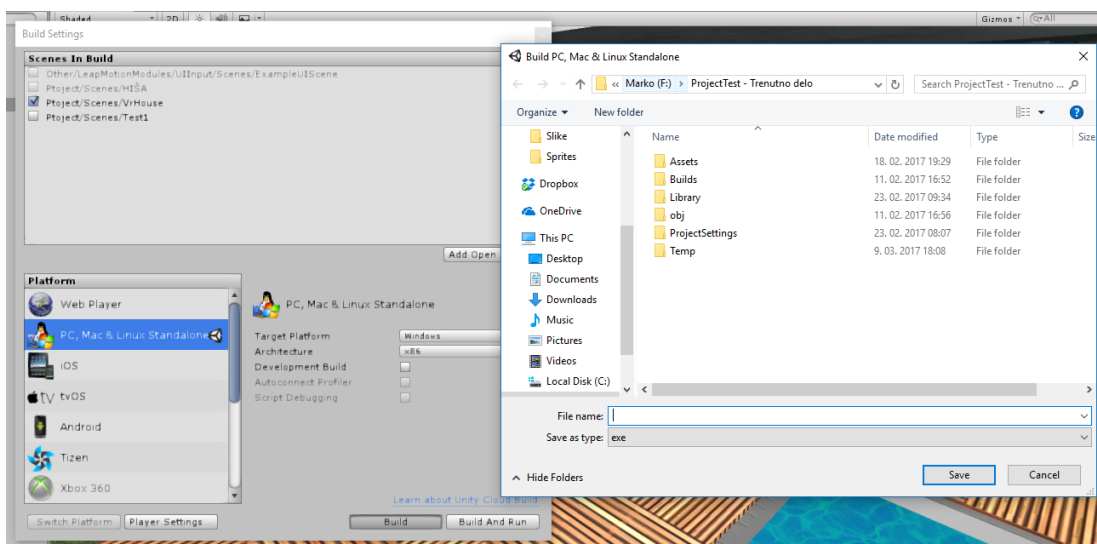
Slika 21: Navodila

4.6 TESTIRANJE

Ko smo napravili verzijo, ki se je dala uporabljati, smo začeli testirati, saj je težave boljše čim hitreje odpraviti. Na koncu bi morali spreminjati večje dele aplikacije. Več ljudi, ki poskusijo aplikacijo, lažje dobimo povratno informacijo, kaj deluje in kaj ne. V programu Unity lahko tudi vidimo FPS (frames per second) in kaj nam vzame največ časa za procesiranje na CPU in GPU (okno Profiler).

4.7 PUBLISH

Pri končnem izvozu aplikacije izberemo začetno sceno, operacijski sistem in kam bomo eksportirali. Lahko pa izberemo resolucijo, dodamo ikono za aplikacijo, na katerih verzijah operacijskega sistema se aplikacija lahko uporablja ter veliko drugih možnosti.



Slika 22: Izvažanje

Na koncu smo aplikacijo izvozili za windows okolje in jo še enkrat testirali. Naložili smo jo na računalnike, na katerih se je kasneje tudi uporabljala.

5 POTRDITEV HIPOTEZ

Na začetku raziskovalne naloge smo si zastavili hipoteze, ki so bile potrjene, vendar smo porabili precej časa in znanja. Pri delu smo naleteli na nekaj težav, a smo jih uspešno rešili.

1. Interaktivna aplikacija služi kot prodajno orodje za komercialista.
 - Naročnik je aplikacijo uspešno uporabil na sejmu MOS (Mednarodni Obrtni Sejem v Celju leta 2016).

(POTRJENA)

2. V navidezni resničnosti se lahko premikam kot v realnosti.
 - Uporabnik se realistično premika do določene meje, saj je nad stranko treba imeti nadzor, da ji lahko usmerjamo pozornost.

(POTRJENA)

3. Aplikacija omogoča interakcijo z objekti.
 - V aplikaciji je možna interakcija in spreminjanje objektov v meniju.

(POTRJENA)

4. Objekti v virtualni resničnosti so zelo realistični in podobni tistim v realnosti.
 - Objekti imajo videz realističnih, a se še vendar zdijo drugačni od pravih, saj se lahko uporabnik z očali za navidezno resničnost zelo približa in ugotovi, da objekt nima toliko detajlov kot pravi.

(POTRJENA)

ZAKLJUČEK

Z izdelkom, ki smo ga ustvarili smo zadovoljni v tako kratkem času. Dobili veliko dobrih in uporabnih izkušenj za nadaljnje projekte.

Veliko stvari bi lahko naredili bolj učinkovito, nekaj stvari bi lahko spremenili in poenostavili. V nadgradnji aplikacije bi lahko vključili posebej narejeno orodje za lažje delo izdelovalca okolja in izdelali avtomatične nastavitve uvoženih slik. Dodali bi lahko tudi več interaktivnih objektov in boljši zvočni dizajn, a to bo naslednji korak pri posodabljanju ali nadgradnji te aplikacije.

Zadovoljni smo tudi, ker odziv uporabnikov, ki so preizkusili aplikacijo dober. Aplikacija je ob prvi predstavitvi pritegnila veliko število ljudi. Mnogo jih še ni slišalo za to tehnologijo zato so jo hoteli preizkusiti. Dosegli smo cilj, da je lahko aplikacija prodajno orodje, s tem pa je prodajalec dobil priložnost razkazati svoje produkte na nov način. Kupci so tako dobili natančnejšo predstavo o vgradnji oken in ostalih materialih. Tako so lažje razumeli tudi komercialista.

VIRI IN LITERATURA

- (1) **Programski jezik C sharp** (spletni vir). 2017. (citirano 9. 3. 2017) Dostopno na naslovu: https://sl.wikipedia.org/wiki/Programski_jezik_C_sharp

- (2) **Unity** (spletni vir). 2017. (9. 3. 2017) Dostopno na naslovu: <https://unity3d.com/>