



UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Izdelava računalniške igre v Unity3D

Poročilo

pri predmetu

Projektni seminar II

Ime in priimek: Patrik Kocjančič

Študijski program: Računalništvo in informatika (magistrski, 2. stopnja)

Koper, 2019

Kazalo

1. UVOD	3
1.1. Namen dokumenta	3
1.2. Opis in razumevanje problema	3
2. SPECIFIKACIJA ZAHTEV	4
2.1. Zahteve	4
2.2. Pregled obstoječih iger	4
3. FUNKCIJSKA SPECIFIKACIJA.....	5
3.1. Splošni opis	5
3.1.1. Perspektiva	5
3.1.2. Uporabniki.....	5
3.2. Funkcionalne zahteve.....	5
4. IZVEDBA	6
4.1. Razvojno okolje	6
4.1.1. Systemske zahteve	6
4.1.2. Namestitvev okolja.....	7
4.2. Gradnja igre.....	7
4.2.1. Dodatki uporabljeni pri igri.....	8
5. IZGLED IZDELANE APLIKACIJE.....	11

1. UVOD

Najbolj igrane igre se v času zelo hitro spreminjajo, kot tudi njihov žanr oz. sistem na kateremu delujejo (sistem z arenami, sistem za preživetje, sistem vsi proti vsem, itd). V zadnjem času so se v svetu igralsva ponovno pojavile igre, ki delujejo na sistemu "battle royale", ki združujejo sistem za preživetje in sistem vsi proti vsem. Igre s tem sistemom so različne, ampak tudi med sabo zelo podobne. Nekatere posnemajo realni svet, nekatere risani oz. izmišljen svet in nekatere tudi kombinacijo med tema dvema. Kljub razlikam vse te želijo privlačiti čim večje število igralcev, zato so funkcije prav tako kot videz zelo pomembna.

1.1. Namen dokumenta

Namen poročila je opisati lastnosti in funkcije računalniške igre, ki je bila izdelana v okviru predmeta Projektni seminar II. Poročilo omenjene aplikacije sestavljajo 4 dokumenti:

- specifikacija zahtev, kjer so opisane zahteve sistema
- funkcijska specifikacija, kjer je opisano zahteve, ki jih mora vsebovati aplikacija
- tehnična specifikacija, kjer je predstavljeno programsko okolje in razni uporabljeni vtičniki pri izdelavi aplikacije
- uporabniška specifikacija, kjer je opisan izgled in uporaba izdelane aplikacije

1.2. Opis in razumevanje problema

Namen projektnega seminarja je zgraditi aplikacijo oz. igro s pomočjo programskega okolja Unity3D. Zgledoval se bom po primerih računalniških iger, ki so v zadnjih letih najbolj igrane (in sicer sistem "battle royale" oz. vsi proti vsem). Cilj projekta se je naučiti programiranja ter modeliranja v 3D okolju v tolikšni meri kot je le to v krajšem času mogoče. Zavedam se da nisem še na nivoju programerjev teh in ostalih uspešnic, zato bom izdelal igro po najboljših zmožnostih s pomočjo pridobljenega znanja.

2. SPECIFIKACIJA ZAHTEV

Izdelava računalniške igre, kljub temu da ne predstavlja končen produkt, ki bo šel na trg, zahteva celovit pristop za rešitev zastavljenega problema. Življenjski cikel igre se nadaljuje s specifikacijo zahtev, ki je predstavljena v naslednjih podpoglavjih.

2.1. Zahteve

Cilj projektnega seminarja je izdelati igro, ki v čim večji meri ustreza izbranemu žanru (sistem "Battle Royale"). Zato so v tem poglavju opisane zahteve oz. želje, ki jih po mojem mnenju mora igra vsebovati:

- osnovne gibe
- grafični vmesnik
- "multiplayer"
- "battle royale" sistem

Osnovni gibi

Igra mora omogočati vse osnovne gibe, katerih prisotnost je izredno pomembna (navigacija in interakcija). Pod osnovne gibe štejemo premikanje po virtualnem okolju, streljanje in ustrezne animacije.

Grafični vmesnik

Grafični vmesnik naj vsebuje vse potrebno in naj bo čim boljše zgrajen. Znotraj igre mora imeti ustrezne napise in prikazovati trenutno zdravje igralca. Če je možno naj vsebuje še ostale komponente, ki jih najnovejše igre vsebujejo.

"Multiplayer"

Igra naj bo zgrajena tako da omogoča igro več igralcev (angl. Multiplayer).

"Battle Royale" sistem

Igra naj deluje na sistemu vsi proti vsem ("Battle Royale" sistem), kjer zmagovalec je tisti ki zadnji ostane v igri.

2.2. Pregled obstoječih iger

V zadnjih letih se je v igralcih ponovno obudil trend igranja iger, ki delujejo na sistemu "Battle royale". Battle Royale igre so igre za več igralcev, ki združuje elemente preživetja, raziskovanja in iskanja elementov iz igre preživetja (angl. survival game) s sistemom zadnjega preživelega. Dober primer tega sistema sta igri *PUBG* in *Fortnite*, ki sta v zelo kratkem času postale zelo priljubljene med igralci. Igrata sta si zelo podobni, saj imata večino enakih funkcionalnosti in tudi njihov stil igranja ter cilj sta enaka. Glavna razlika med igrata je seveda tip igre, saj prva čim bolj posnema realni svet, medtem ko druga je bolj animirana. Ker sta igri zelo priljubljeni, sem se odločil da bi zgradil igro, ki posnema omenjeni uspešnici.

3. FUNKCIJSKA SPECIFIKACIJA

Namen funkcijske specifikacije je opisati lastnosti in funkcije izdelane računalniške igre.

3.1. Splošni opis

3.1.1. Perspektiva

Predstavljena igra ne bo objavljena, ampak bo izdelana le v okviru učenja uporabe programskega orodja Unity3D za predmet Projektni seminar II. Igro bo preizkusila le peščica ljudi v okviru predstavitve projekta pri predmetu, saj ta predstavlja prve korake v učenju programiranja računalniških aplikacij oz. iger v Unity3D.

3.1.2. Uporabniki

Predstavljena igra ne bo namenjena širši javnosti, saj je namenjena le za učne namene programerja. Kljub temu bo uporaba omogočena vsem, ki jo bodo želeli preizkusiti.

3.2. Funkcionalne zahteve

V tem poglavju so opisane funkcionalne zahteve glavnega menija in dejanske igre.

Funkcionalne zahteve glavnega menija igre so sledeče:

- zaženi igro
 - host igra
 - client igra
- zbiranje različnih orožij
- izhod iz aplikacije

Zaženi igro

Uporabnik lahko zažene igro in pri tem lahko izbira ali bo "host" oz. gostitelj, ki predstavlja server igre na kateremu se odvija glavni del igre ali bo "client" oz. odjemalec kateri pošilja povpraševanja in prejema odgovore za različne storitve (vse kar hoče "client" uporabnik naredi mora poslati serverju oz. "host"-u da lahko ta pošlje vsem ostalim igralcem). Če je uporabnik prvi igralec ki dostopa do igre bo imel le možnost kreiranja igre in tako bo postal gostitelj le-te. Vsi ostali pa imajo možnost kreiranja ali pridruževanja k igri gostiteljev.

Zbiranje različnih orožij

Naslednja možnost med katerimi lahko uporabnik izbira je "Options", kjer lahko uporabnik izbere željena orožja za igro.

Izhod iz aplikacije

Zadnja možnost med katerimi lahko uporabnik zbira je "Exit", preko katere imamo možnost izhoda iz aplikacije.

Funkcionalne zahteve igre so sledeče:

- "Pause menu"

“Pause menu”

Pri izvajanju igre lahko uporabnik odpre “Pause menu” s klikom na “escape” tipko, kjer lahko izbira z nadaljevanjem igre ali z izhodom iz igre. S klikom na zadnjo se uporabnik vrne na glavni meni.

4. IZVEDBA

4.1. Razvojno okolje

Predstavljena igra je izdelana s pomočjo programskega okolja Unity3D. Unity je večplatformni igralni motor, ki ga je razvilo in prvič objavilo podjetje *Unity Technologies* leta 2005. Leta 2018 so igralni motor Unity razširili na več kot 25 platform. Okolje unity je namenjeno kreiranju tridimenzionalnih, dvodimenzionalnih, VR in AR iger, kot tudi simulacij in ostalih izkušenj. Unity uporabljajo tudi v različnih industrijah med katerimi filmska, avtomobilska, itd. C# je primarni programski jezik, ki se uporablja pri programiranju v Unity.

Poleg Unity sem za delo uporabil tudi Adobe Fuse in spletno aplikacijo Mixamo. Adobe Fuse omogoča zelo poenostavljeno kreacijo avatarjev, saj ni potrebno biti izkušen 3d modeler za hitro kreacijo in preoblikovanje le teh. Iz Adobe Fuse lahko z klikom na eno tipko prenesemo model v spletno aplikacijo Mixamo. Mixamo omogoča dodajanje skeleta avatarju in nato dodajanje različnih animacij ki jih lahko kasneje vse skupaj prenesemo na računalnik in vstavimo v Unity.

4.1.1. Sistemske zahteve

V tem podpoglavju so predstavljene sistemske zahteve za vsako uporabljeno orodje v okviru razvoja.

Glavne sistemske zahteve za razvoj v Unity so naslednje:

- **OS:** Windows 7 SP1+, 8, 10, 64-bit versions only; Mac OS X 10.9+.
- **CPU:** Podpora za SSE2
- **GPU:** Grafična kartica z DX9 (“shader” model 3.0) ali DX11 z zmogljivostjo 9.3.
 - NVIDIA GeForce GTX 780 ali AMD Radeon R9 290X (3 GB VRAM) ali boljše
- **RAM:** 8GB
- Ostalo je večinoma odvisno od zahtevnosti in potreb projektov.

Glavne sistem zahteve za Adobe Fuse so naslednje:

- **OS:** Windows 7, 8, 10; Mac OS X v10.8 - v10.11
- **CPU:** Intel Pentium 4 2.00GHz ali boljši
- **RAM:** 8GB
- **GPU:** Podpora OpenGL 2.2+

Glede na to da je Mixamo spletna aplikacija potrebuje le spletni brskalnik, kateri podpira WebGL. Spodaj naštetih so priporočeni brskalniki:

- Microsoft Internet Explorer 11; Edge 12, 13
- Firefox 38+

- Chrome 31+
- Safari 8+

4.1.2. Namestitev okolja

Namestitev okolja je precej enostavna, saj je potrebno le prenesti zadnjo različico iz uradne spletne strani in nato namestiti na računalnik. Pri namestitvi se lahko uporabnik oziroma v tem primeru lahko tudi programer odloči katere dodatne komponente bo potreboval za programiranje. Komponente med katerimi lahko zbiramo so podpore za različne sisteme (Windows, Android, Mac, itd.) in tudi ustrezna različica Visual Studio, ki je potreben programiranje s skriptami. Po namestitvi lahko uporabnik zažene Unity in začne s kreiranjem svojih projektov.

Unity omogoča tudi dodajanje različnih dodatkov, vtičnikov (angl. plug-in) in drugih sredstev, ki jih lahko uporabnik pridobi iz Unity-jeve trgovine sredstev (Asset Store), ali pa tudi iz drugih virov. Ta sredstva omogočajo hitro personalizacijo projekta brez izgube časa in ogromno programiranja.

Za delo sem potreboval tudi program Adobe Fuse, ki sem ga prav tako prenesel in namestil z privzetimi nastavitvami.

4.2. Gradnja igre

Gradnjo igre smo najprej začeli z kreiranjem projekta, ki sem ga imenoval kar "ShooterGame". Na začetku projekta sem najprej kreiral teren na katerem se bo igra odvijala. Za boljši izgled terena sem uporabil vgrajeno orodje znotraj Unity, ki sprejema višinsko sliko in ustrezno preuredi teren le-tej. Da bi izgled teren bil čim bolj podoben realnemu sem se odločil za spletno stran terrain.party (<http://terrain.party/>), kjer lahko gost enostavno izbere poljubno površino med 64km² in vse do 3600km² ter nato prenese višinsko sliko izbranega področja. V Unity sem to površino strnil teren na veliko manjše področje z ustreznimi spremembami, saj bi to bilo preveliko območje igranja. Pri tem sem poskušal ohraniti originalno konfiguracijo terena. Po končanem terenu sem se lotil kreacije in animacije avatarja za igro s pomočjo Adobe Fuse in Mixamo. Odločil sem se za uporabo teh, ker je ta proces izredno zahteven in časovno potraten za začetnike. V Adobe Fuse sem sestavil avatar in nato sem vsak dela telesa lahko ustrezno spreminjal. Končan avatar sem prenesel v Mixamo, kjer je ta dobil najprej skelet, saj brez tega ne moramo dodajati animacij le-temu. Po dodajanju skeleta sem prenesel avatar v Unity in začel z iskanjem primernih animacij iz zbirke Mixamo. Ustrezno animacijo sem lahko še po potrebi spremenil, toliko kot je le bilo to možno in nato prenesel tudi te v Unity. Kljub temu da so animacije že bile narejene in da ima Unity komponento Animator, kjer lahko urejamo animacije, je bilo potrebno veliko dela pri usklajevanju gibov in prehodu med različnimi gibi. Za enostavne gibe v različnih pozah (brez, puška, pištola) je bilo potrebno več kot 50 animacij. Ker animacije niso bile popolne sem dodal še inverzno kinematiko, da sem uredil določen gib izven animacije. Inverzna kinematika omogoča, da lahko programer popolnoma spremeni gib dela telesa avatarja v animaciji. To pomeni da bo avatar naredil ta gib ki je določen inverzni kinematik kljub temu da se animacija predvaja s popolnoma različnimi gibi. Naslednji korak je bil sprememba igre iz eno igralne igre v več igralno igro z že vgrajenim dodatkom za povezovanje preko omrežja. Pred tem je bilo potrebno tudi ta dodatek ("Unet") omogočiti preko servisne strani, ki je dostopna kar iz projekta. Nato je sledilo programiranje raznih interakcij med različnimi igralci, ki so se morale preko serverja prenesti na vsakega od igralcev. Ta del je bil najtežji del gradnje igre, saj je moral vsak igralec najprej

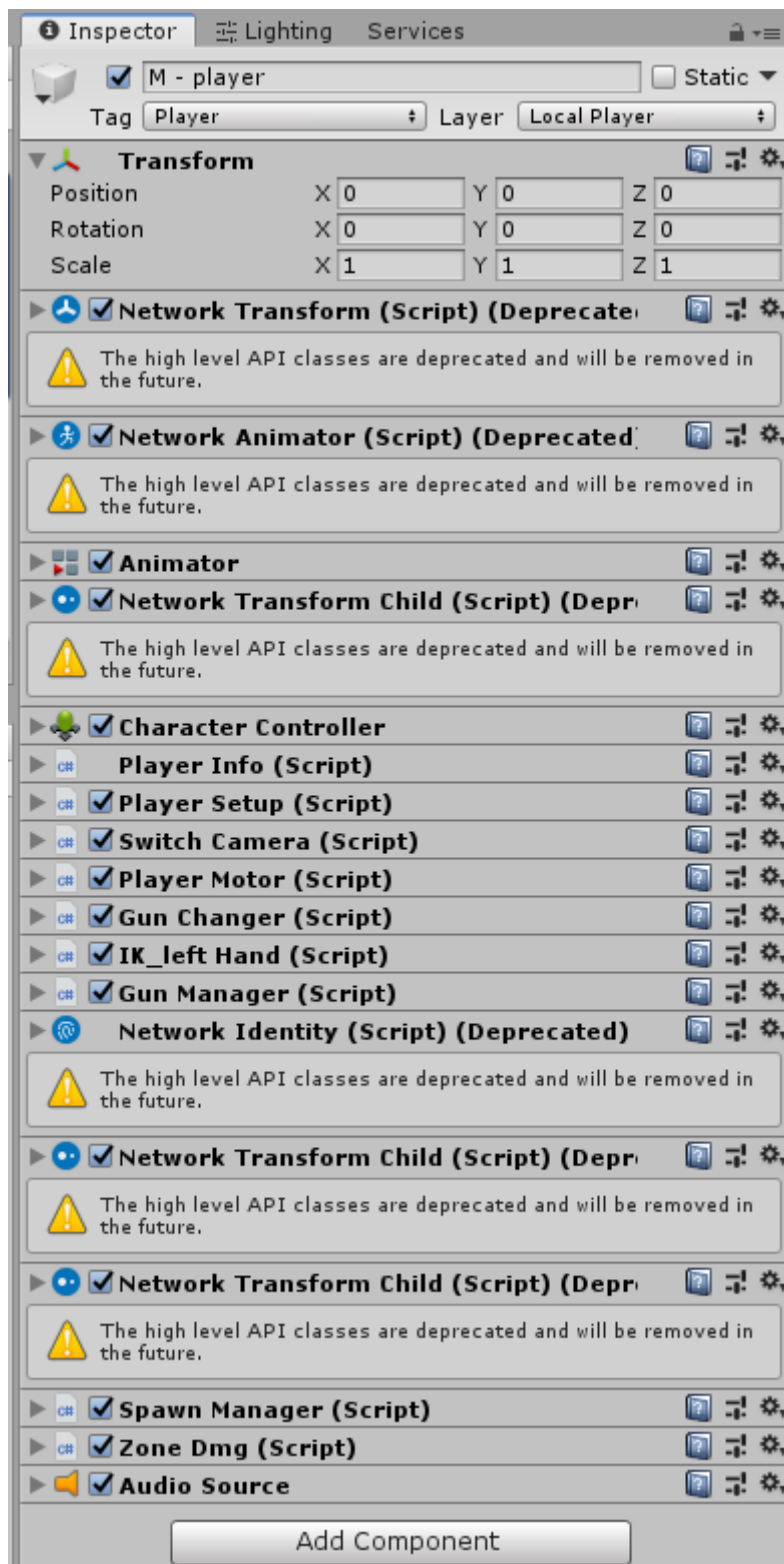
vprašati za določeno akcijo, ta akcija obdelal na serverju in šele nato je server lahko prenesel to akcijo na vse igralce. Na koncu sem še dopolnil igralno površino z ustreznimi funkcionalnostmi in z cono, ki zmanjšuje igralno površino.

4.2.1. Dodatki uporabljeni pri igri

Kot že rečeno sem pri gradnji igre uporabil že vgrajen dodatek , kateri omogoča igro več igralcev, to je Unity Multiplayer ("Unet"). Uporabil sem tudi nekaj sredstev iz Asset Store.

Unity Multiplayer ("Unet")

Unet je dodatek, ki omogoča programerju enostavno kreacijo več igralških iger, saj ta vsebuje vse potrebne skripte in komponente za različne objekte za delo preko omrežja. Vsak objekt ki se mora osveževati pri vseh igralcih mora imeti eno ali več komponent, ki jih ponuja Unet. Glavna komponenta, ki jo mora imeti objekt za osveževanje oz. da ga lahko server sploh spozna je omrežna identiteta (angl. Network Identity). Brez te se ne bodo akcije osveževale preko omrežja. Izbira komponent je odvisna od akcij ki jih želimo osvežiti preko omrežja. Na primer na avatarju nimamo le ene ali dveh komponent, ki ji Unet ponuja, temveč jih imamo kar šest če štejemo tudi identiteto. Vsi so izredno pomembni za izvajanje celotne igre. Medtem ko najpomembnejša sta omrežna transformacija (angl. Network Transform) in omrežni animator (angl. Network Animator), saj brez prve se avatar ne bi mogel premikati in brez druge bi avatar bil zmeraj v enaki pozi. Slika 1 prikazuje vse komponente, ki jih avatar ima na sebi. Z vidika programerjev je Unet že precej star, saj se tehnologija na tem področju neprestano spreminja in nadgrajuje in zaradi tega bo ta tudi v kratkem prenehal delovati v prid novega dodatka, ki je tudi že v uporabi in je predstavlja prihodnost več igralškega programiranja v Unity (Photon).

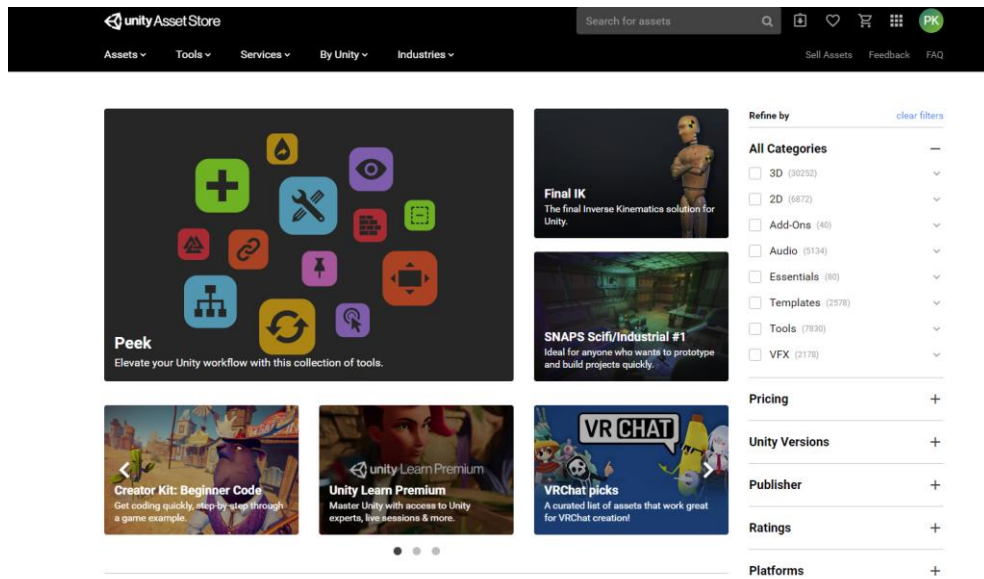


Slika 1: Set komponent na vsakem avatarju

Asset Store

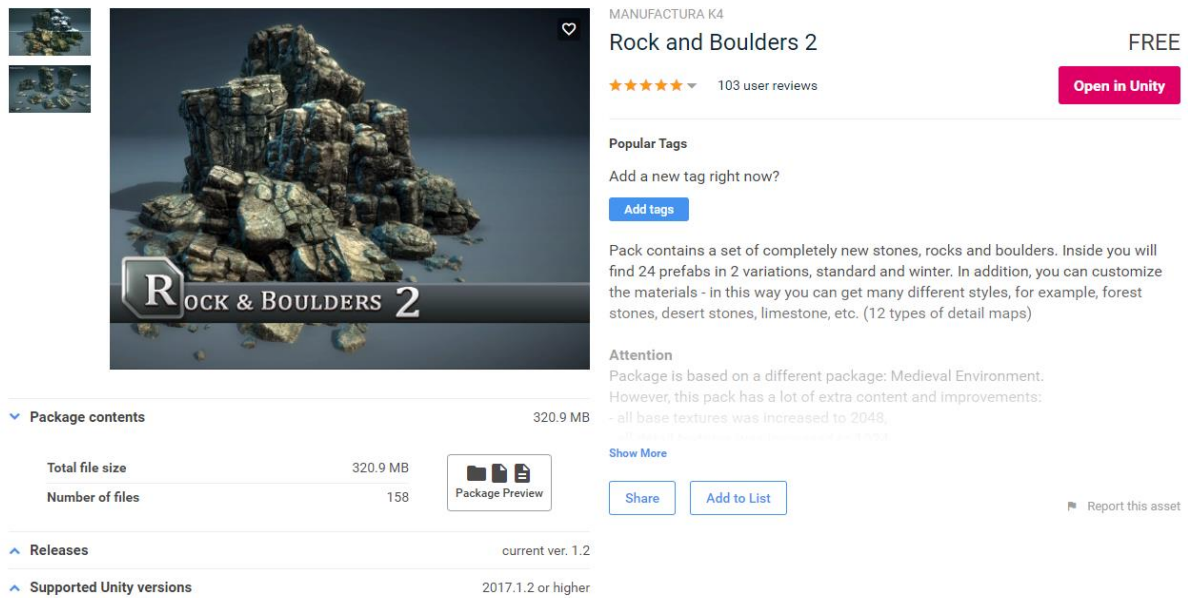
Asset Store je trgovina sredstev, v kateri lahko programer pridobi različne dodatke, vtičnike in sredstva. Slika 2 prikazuje Asset Store. Iz trgovine lahko programer prenese različne elemente, ki so potrebni v njegovem projektu in za katere bi v nasprotnem primeru potreboval izredne količine časa za programiranje in modeliranje. V trgovini najdemo plačljive, ampak tudi

izdelke, ki so zastonj in so primerni za vse bodoče programerjem za testiranje in učenje, preden se odločijo za programiranje večjih in resnih projektov, ki bodo objavljeni.



Slika 2: Unity Trgovina dodatkov, vtičnikov in sredstev (Unity Asset Store)

Iz trgovine sem prenesel nekaj paketov za izboljšanje izgleda terena. Med temi sem prenesel paket 3D kamnov in skal, ki sem jim raztresel po celotni površini terena in je prikazan na Slika 3.



Slika 3: Paket 3D kamnov in skal

5. IZGLED IZDELANE APLIKACIJE

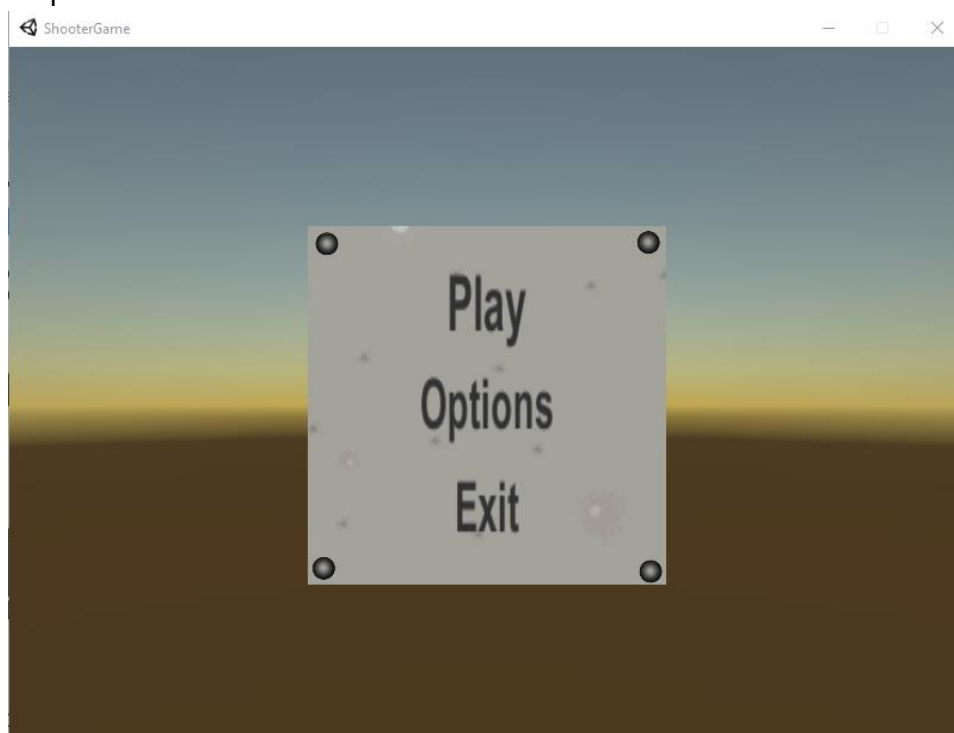
V tem poglavju bom predstavil izgled izdelane aplikacije. Poskušal sem narediti oz. implementirati vse osnovne funkcije in večino ostalih funkcionalnosti sodobnih strelskih iger. Med osnovne funkcije štejemo premikanje, več igralški sistem, interakcijo z aktivnimi objekti kot so drugi igralci, orožja in področje igranja (področje igranja se s časom zmanjšuje) ter interakcijo z pasivnimi objekti kot je teren z vsemi pod objekti. Medtem ko med ostale funkcionalnosti štejemo na primer razne zvoke, glavni meni in vse ostalo, ki ni ključnega pomena za igranje igre.



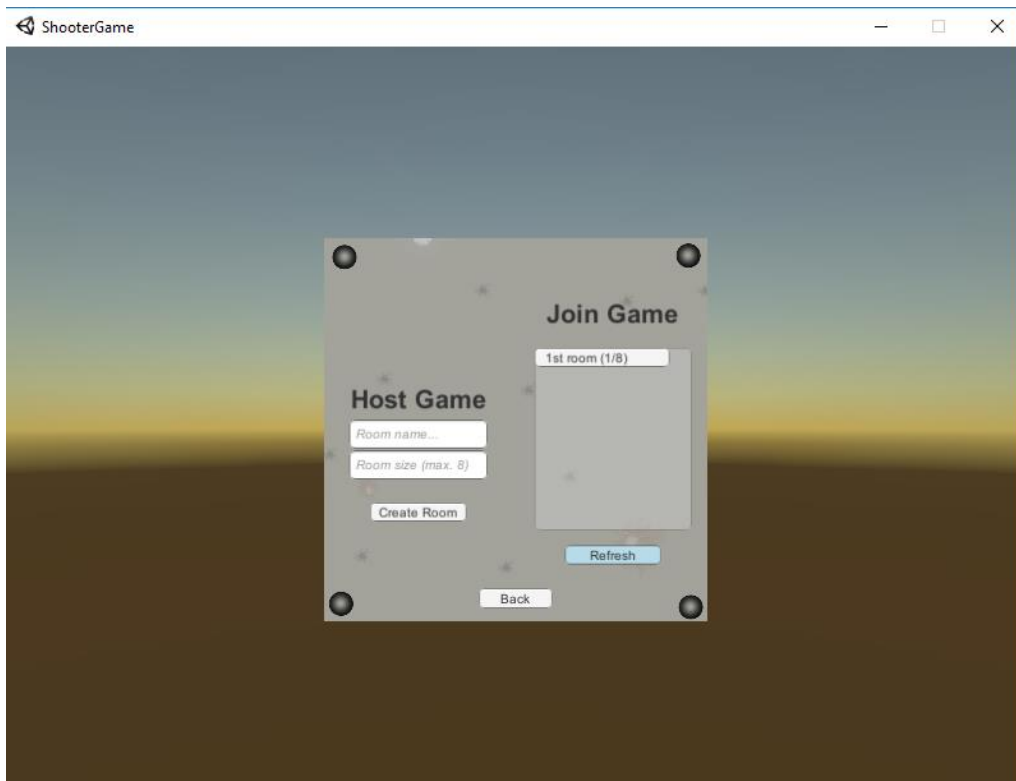
ShooterGame.exe

Slika 4: Ikona za zagon aplikacije

Z dvoklikom na ikono aplikacije (Slika 4) uporabnik prispe do glavnega menija aplikacije. Na glavnem meniju lahko uporabnik izbira med 3 možnostmi. Prva možnost je "Play" preko katere uporabnik dostopa do menija za igro. Druga možnost je "Options", kjer lahko trenutno uporabnik nastavlja določene parametre in še ni dokončno implementirana. Prva in druga možnost bodo opisane v nadaljevanju. Tretja in zadnja možnost je "Exit", katere kot že ime pove omogoča uporabniku zaprtje oz. izhod iz aplikacije. Na Slika 5 je prikazan glavni meni z predhodno opisanimi možnostmi.

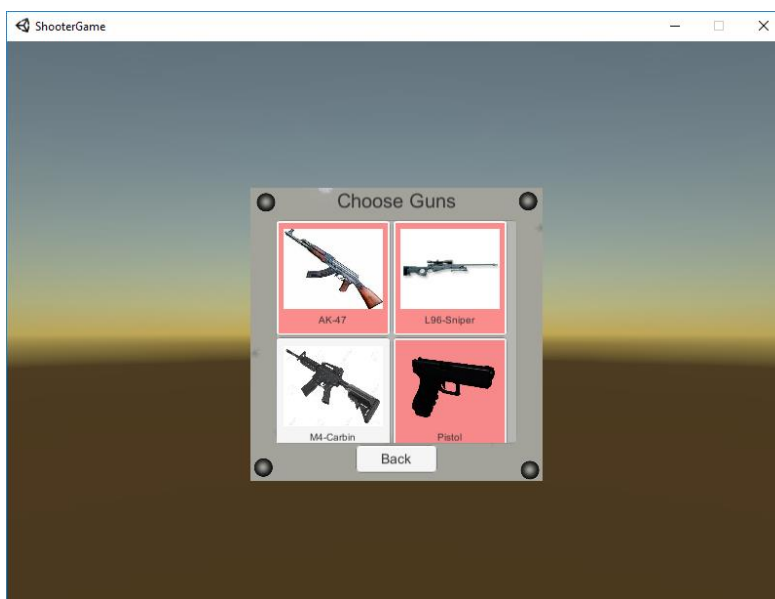


Slika 5: Glavni meni aplikacije



Slika 6: Meni igre ("Play Menu")

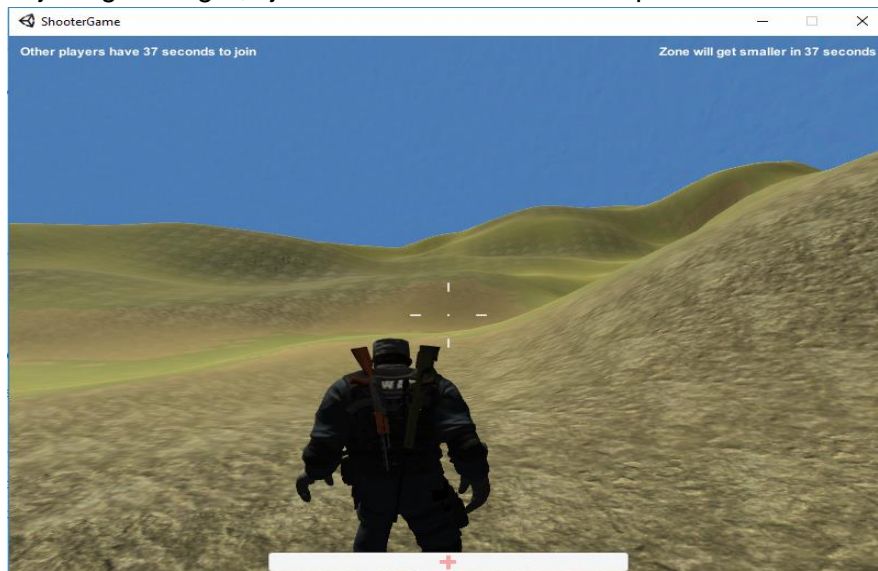
Slika 6 prikazuje naslednji meni, to je meni igre (angl. Play Menu), ki je razdeljen na dve polovici. Na levi polovici ima uporabnik možnost kreirati svojo sobo za igro, kateri lahko izbere ime in število igralcev (min: 2, max: 8), ki lahko vstopi v to. Uporabnik po vpisanih parametrih lahko kreira sobo z gumbom "Create Room". V primeru da uporabnik ne vpiše številke bo kreirana soba za maksimalno število oseb, ki je 8. Na desni polovici ima seznam aktivnih in nezapolnjenih sob na katere s klikom na njih se lahko pridruži. Pod tem pa je gumb "Refresh" s katerim lahko zgornji seznam osveži z novo nastalimi sobami. Na koncu pa je še gumb "Back" s katerim se uporabnik vrne na glavni meni.



Slika 7: Nastavitveni meni ("Option Menu")

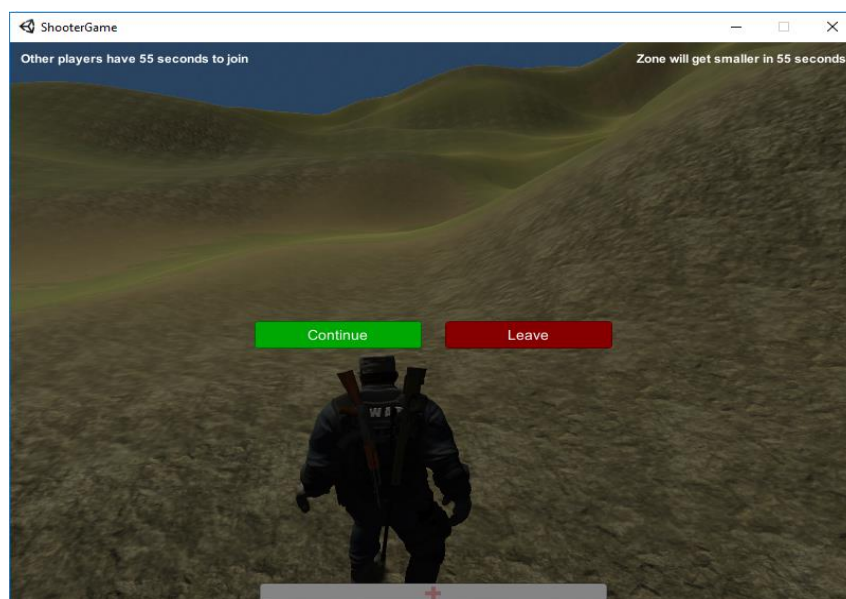
Sledi nastavitveni meni (angl. Options Menu), ki je prikazan na Slika 7. Pri tem meniju lahko izberemo poljubno kombinacijo orožij, s katerimi bo avatar v igri opremljen. Orožja izberemo s klikom, tako je to označeno z rdečo in na koncu ko imamo izbrana orožja uporabnik lahko zapusti nastavitveni meni in začne z igro. Trenutno funkcija za opremljanje avatarja ni še implementirana, zato bo avatar ob začetku igre opremljen s privzetimi orožji. Set privzetih orožij je tudi označen na Slika 7.

To je bilo vse kar se tiče glavnega menija aplikacije, zato se bomo preselili v dejansko igro. Slika 8 prikazuje zagnano igro, kjer lahko vidimo avatar v 3d prostoru.



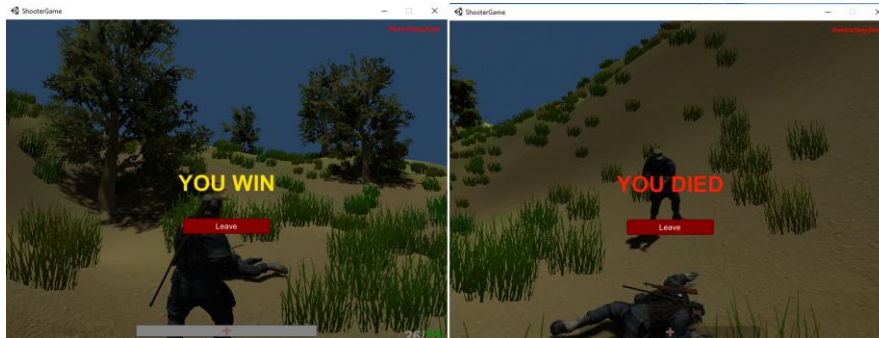
Slika 8: Primer igralca v igri

V teku igre lahko uporabnik klikne na tipko "Escape", ki odpre meni znotraj igre, ki je prikazan na Slika 9. Na tem meniju lahko uporabnik izbira med dvema možnostma, in sicer med "Continue" za nadaljevanje igre in med "Leave" za izhod iz igre in vrnitev na meni igre.



Slika 9: Meni znotraj igre, kjer lahko uporabnik izbira med nadaljevanjem ali izhodom iz igre

Na koncu imamo še Slika 10, na kateri je na levi prikazan zaslon zmagovalca in na desni zaslon poraženca, katere se pokažejo določenim igralcem glede na njihovo uspešnost v igri. Zaslon zmagovalca se prikaže le zadnjemu igralcu v sobi, in ostalim seveda zaslon poraženca. Pri obeh zaslonih se pojavi tudi gumb "Leave" tako da se vsak uporabnik lahko vrne na meni igre.



Slika 10: Primer zmagovalca (levo) in poraženca (desno)