

# Razvoj Virtual Reality računalne igre u Unity Engineu

---

**Vuković, Marko**

**Master's thesis / Diplomski rad**

**2021**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka / Sveučilište u Rijeci**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:195:514844>

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

*Download date / Datum preuzimanja:* **2024-07-17**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



**SVEUČILIŠTE U RIJECI  
ODJEL ZA INFORMATIKU  
Diplomski studij informatike – informacijski i komunikacijski sustavi**

**Marko Vuković  
Razvoj Virtual Reality računalne igre u Unity Engineu  
Diplomski rad**

**Mentor: izv. prof. dr. sc. Marina Ivašić-Kos**

**Rijeka, Rujan 2021.**

# Sadržaj

1. Sažetak.....	1
2. Uvod.....	2
2.1 Inspiracija .....	6
3. Alati za razvoj igara.....	7
3.1 Unity.....	7
3.2 Valve Index.....	8
4. Opis igre.....	9
4.1 Korišteni asseti.....	9
4.1.1 VRIF.....	9
4.1.2 Gaia 2 – Terrain Generator .....	10
4.1.3 Emerald AI .....	11
4.1.4 Toy Soldiers .....	12
5. Izrada VR igre .....	13
5.1 Podešavanje Unity Editora .....	13
5.2 Kreiranje levela.....	17
5.3 Kreiranje glavnog lika .....	22
5.4 Kreiranje AI neprijatelja .....	30
5.5 Kreiranje glavnog izbornika .....	37
5.6 Kreiranje Game Over scene .....	38
6. Zaključak.....	39
7. Literatura.....	40
8. Popis slika.....	42
9. Prilozi .....	44

Rijeka, 13.2.2021.

## **Zadatak za diplomski rad**

Pristupnik: **Marko Vuković**

Naziv diplomskog rada: **Razvoj Virtual Reality računalne igre u Unity Engineu**

Naziv diplomskog rada na eng. jeziku: **Development of VR video game in Unity Engine**

Sadržaj zadatka:

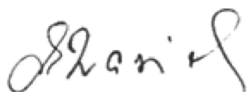
Opisati osnovne koncepte virtualne stvarnosti te kratak razvoj i karakteristike uređaja koji se koriste za generiranje i prikaz 3D modela. Proučiti vrste računalnih igara i kratko ih opisati. Istražiti tehnologiju i alate koji se koriste za izradu VR igre te ukratko opisati ključne karakteristike alata koji će se koristiti u praktičnom dijelu rada.

Osmisliti igru proizvoljnog žanra za jednog igrača i izraditi je u VR okruženju. Opisati potrebne prilagodbe u Unity engineu za izradu 3D VR igre te povezivanje s virtualnim naočala za igranje.

Opisati dizajn i mehanike igre te objasniti ključne korake razvoja VR igre kao što je izrada 3D modela, teksturiranje površina, kreiranje glavnog lika i neprijatelja, izrada prepreka, izrada animacija, upravljanje kretanjem, kreiranje UI izbornika, računanje napretka u igri i slično.

Mentor:

Izv. prof. dr. sc. Marina Ivašić-Kos



Voditeljica za diplomske radove:

Izv. prof. dr. sc. Ana Meštrović



Zadatak preuzet: 25.02.2021

*Marko Vuković*

(potpis pristupnika)

## 1. Sažetak

U ovom radu prikazani su postupci izrade VR (*eng. Virtual Reality*) igre u Unity Engine-u. Također su objašnjeni svi asseti koji su se koristili za njezin razvoj, način podešavanja Unity Engine-a te objašnjenje rada virtualnih naočala Valve Index koje su se koristile za testiranje igre.

Na početku rada je opisan Unity Engine kao program za izradu videoigara, njegove bitne značajke te povijest njegovog razvoja. Sljedeće je detaljan opis stvaranja igre u VR okruženju u obliku uputa tako da bilo tko može napraviti igru na ovaj način. Igra je rađena za jednog igrača i mora se igrati pomoću virtualnih naočala na računalu. Spada pod „sandbox“ (*hrv. pješčanik*) žanr igara, što znači da je to mješavina više različitih žanrova. U ovom slučaju je to mješavina pucačine, avanturističke i „survival“ (*hrv. preživljavanje*) igre

Ključne riječi: Unity Engine, VR, HMD, Valve Index, assets, Emerald AI, Gaia, VRIF, Prefab, objekt

## 2. Uvod

Virtualna stvarnost (dalje VR) je uporaba računalnog modeliranja i simulacije koja omogućuje osobi interakciju s umjetnim trodimenzionalnim (3-D) vizualnim ili drugim osjetilnim okruženjem [1]. VR aplikacije uranjaju korisnika u računalno generirano okruženje koje simulira stvarnost pomoću interaktivnih uređaja koji šalju i primaju informacije te se nose kao naočale, slušalice, rukavice ili odijela za cijelo tijelo [2]. U tipičnom VR formatu, korisnik koji nosi kacigu sa stereoskopskim zaslonom gleda animirane slike simuliranog okruženja. Iluziju da je osoba stvarno tamo stvaraju senzori pokreta koji hvataju korisnikove pokrete i sukladno tome prilagođavaju prikaz na zaslonu, uglavnom u stvarnom vremenu (u trenutku kad se kretanje korisnika dogodi). Tako korisnik može obići simuliranu instancu prostora, iskusiti različite pozicije gledišta i perspektiva koje su uvjerljivo povezane s vlastitim okretanjima glave i hodom. Noseći umjetne rukavice opremljene sensorima za imitaciju sile koji pružaju osjećaj dodira, korisnik može čak pokupiti i manipulirati objektima koje vidi u virtualnom okruženju [3].

Trenutni životni ciklus virtualne stvarnosti započeo je kada su se prvi prototipovi virtualnih naočala „Oculus Rift“ pojavili na sajmu videoigara E3 2012. godine [4], ali njezin začetak je bio još u prvoj polovici 20. stoljeća. Ideja o uranjanju u 3D okruženja seže sve do stereoskopa koji su plijenili maštu ljudi u 19. stoljeću. Ako se svakom oku predstavi gotovo identična slika, ljudski mozak će ih kombinirati i pronaći dubinu u njihovim razlikama. To je isti mehanizam koji je poznati stereoskop „View-Master“, Slika 1, koristio i tako postao slavan [5].



Slika 1: View Master

Razvoj VR-a doživio je vrhunce i padove. Prvi sustav VR ekrana na glavi (*eng. Head Mounted Display, HMD*) zvao se Damoklov mač, Slika 2. Izumili su ga računalni znanstvenik Ivan Sutherland i njegov student Bob Sproull 1968. godine [6]. U međuvremenu je izraz "virtualna stvarnost" popularizirao Jaron Lanier 1980-ih [2]. Deset godina kasnije, VR se koristio za obuku i simulaciju u američkoj vojsci i Nacionalnoj upravi za aeronautiku i svemir (NASA) [2]. Masovna proizvodnja VR sustava započela je početkom 1990.-ih predvođena tvrtkama Virtuality koja je stvarala namjenske VR arkadne sustave i Sega koja je stvorila VR sustav za svoju „Mega Drive“ konzolu [7, 8]. Danas postoji više poznatih VR HMD-a, a neki od najpoznatijih su: Oculus Rift tvrtke Facebook koji ima više verzija, Playstation VR tvrtke Sony za Playstation 4 konzolu, Vive HMD koji također dolazi u više verzija tvrtke HTC te Valve Index tvrtke Valve koji je trenutno najnoviji na tržištu. Također postoje HMD-ovi za mobilne uređaje kao što su Google Cardboard i Samsung Gear VR [9].



Slika 2: Damoklov mač

Oculus Rift, Slika 3, je prekinuta linija HMD-a za VR koju je razvio i proizveo Oculus VR, koji je sada dio Facebooka, a objavljen je 28. ožujka 2016. godine. Rift je prošao kroz razne prototipske modele od Kickstarter kampanje, od kojih je oko pet predstavljeno javnosti prije nego što je stiglo do komercijalnog izdanja. Dva od ovih modela isporučena su testerima označena kao razvojni kompleti DK1 sredinom 2013. godine i DK2 sredinom 2014. godine. Oculus trenutno nudi samo Oculus Quest 2 HMD. To je jedini HMD koji sadrži integrirani mobilni računalni hardver i ne zahtijeva računalo ili konzolu za rad. Pušten je u prodaju u rujnu 2020. godine te je nasljednik Oculus Rift HMD-a [10].



**Slika 3: Oculus Rift**

PlayStation VR, Slika 4, je HMD koji je razvila tvrtka Sony Interactive Entertainment, a objavljene su u listopadu 2016. godine. Od 31. prosinca 2019. godine PlayStation VR je prodao 5 milijuna jedinica diljem svijeta te ga to čini jednim od najprodavanijih VR HMD-a. Trenutno, to je jedini headset koji se koristi na konzolama, točnije na Playstation 4 i Playstation 5 konzoli [11].



**Slika 4: Playstation VR**

HTC Vive, Slika 5, je headset koji su razvili HTC i Valve. Headset koristi tehnologiju praćenja pomoću eksternih senzora omogućujući korisniku da se kreće u 3D prostoru i koristi ručne kontrolere s praćenjem kretanja za interakciju s okolinom. Postoji više verzija Vive headseta, a to su: Vive Pro, Vive Pro Eye, Vive Cosmos i Vive Pro 2. Vive Cosmos je jedini headset tvrtke HTC kojem nisu potrebni eksterni senzori za praćenje pokreta. Umjesto njih koristi 4 ugrađene kamere koje prate pokrete igrača. Ostale verzije headseta su vrlo slične, razlika je uglavnom u kvaliteti rezolucije panela gdje



Vive Pro 2 ima najveću rezoluciju od 2448\*2448 piksela po panelu, a Vive Pro i Pro Eye imaju rezoluciju od 1440\*1600 piksela po panelu [12].



Slika 5: HTC Vive

Valve Index, Slika 6, je VR headset koji je stvorila i proizvela tvrtka Valve. Headset je najavljen 30. travnja 2019. godine i objavljen je 28. lipnja 2019 godine. Index je HMD druge generacije i trenutno jedini headset koji proizvodi Valve. Koristi eksterne senzore za praćenje pokreta koji su zapravo nadograđena verzija Vive eksternih senzora što znači da su međusobno kompatibilni. Novitet kod ovog headseta su Valve Index kontroleri, poznatiji pod imenom „Knuckles“, koji imaju 87 senzora za praćenje položaja ruke, kretanja i stiskanja šake kako bi se stvorio točan prikaz ruke korisnika u virtualnom okruženju [13].



Slika 6: Valve Index

## 2.1 Inspiracija

Inspiracija za igru bila je igra Hot Dogs, Horseshoes & Hand Grenades ili skraćeno „H3VR“, Slika 7, od proizvođača Rust LTD. [14]. Žanr igre je „shooting sandbox“ što znači da ne postoji određen cilj u igri, nego je stvorena za zabavu i testiranje različitih oružja i oruđa. Igra je poznata po najvećem broju raznolikih vatrenih oružja od bilo koje druge VR igre i za svako vatreno oružje prikazano je kako funkcionira u stvarnom svijetu. Sadrži mnogo različitih scenarija, od običnih streljana i vojnih poligona do avantura i natjecateljskih izazova [15]. H3VR je 2020. godine osvojila „Platinum“ nagradu u kategoriji najboljih VR igara na Steam platformi što je najviša moguća nagrada na toj platformi [16]. Na slici 8 prikazana je scena iz igre.



Slika 7: H3VR



Slika 8: Scena iz igre

### 3. Alati za razvoj igara

Početakom 90-ih godina počeo se koristiti termin „game engine“. Game engine je softverska okolina (engl. Framework) prvenstveno osmišljen za razvoj videoigara i općenito uključuje relevantne knjižnice i programe podrške. Programeri mogu koristiti game engine za izradu igara za računala i konzole. Osnovna funkcionalnost koju tipično pruža game engine može uključivati mehanizam za renderiranje 2D ili 3D grafike, mehanizam za fiziku ili otkrivanje sudara, zvuk, skriptiranje, animaciju, umjetnu inteligenciju, umrežavanje, upravljanje memorijom, podrška za lokalizaciju, itd. [17]. Jedan od najpoznatijih je „Unreal Engine“ koji je objavljen 1998. godine i korišten u igri „Unreal“ [18]. Nakon Unreal Enginea, nastalo je još nekoliko game enginea kao npr. Unity, RPG Maker, Godot, CryEngine, GameMaker itd. U ovom radu korišten je Unity game engine.

#### 3.1 Unity

Unity, Slika 9, je cross-platform (hrv. *više-platformski*) game engine koji je stvorila tvrtka Unity Technologies u lipnju 2005. godine. Na početku je bio rađen samo za Mac OSX, no sada se proširio na više od 25 različitih platformi [19]. Neke od glavnih značajki Unity-a su [20]:

- Koristi C# programski jezik te objektno-orijentirani pristup programiranju naspram Unreal Enginea gdje se koristi C ++ programski jezik i proceduralni pristup [21].
- Podržava 3D i 2D grafiku s različitim funkcionalnostima koji su potrebni u raznim žanrovima igara.
- Implementiran je navigacijski sustav za AI (hrv. *umjetna inteligencija*) pomoću kojeg se može kreirati NPC (eng. *non-playable character*) koji se može snalaziti u virtualnom svijetu.
- Sadrži „Asset store“ preko koje se mogu kupiti različiti dodaci, gotovi modeli, alati i ostali resursi koje su pravili amateri, ili čak ozbiljnije kompanije koje se izričito bave modeliranjem.
- Prefabs, tj. gotovi i uređeni modeli koji su spremni za korištenje u vlastitim Unity projektima. Svaki novi model koji je stvoren u Unityu ili već postojeći model s promijenjenim postavkama može se spremirati kao Prefab te će tako sačuvati sve izmjenjene postavke na njemu.
- Ugrađen UI (eng. *User interface*) sustav koji omogućuje jednostavnu implementaciju korisničkog sučelja.

Unity 2 s 50 novih dodataka [19] izašao je 2007. godine. Neki od njih su npr. mogućnost kreiranja dinamičnih sjena u stvarnom vremenu i poboljšani terrain

engine za 3D okolinu. Uz sve te novitete došla je i mnogo tražena podrška za stvaranje multiplayer igara, što je znatno povećalo popularnost Unity Engine-a.

Unity 3 s dodanom podrškom za Android sustave izašao je 2010. godine. Također, uz Android podršku, dodana su poboljšanja grafičkih značajki za stolna računala i konzole [19].

Unity 5 izašao je 2015. godine s poboljšanim postavkama osvjetljenja i zvuka, boljom kompatibilnosti s Web preglednicima i novim Nvidia PhysX3.3 engineom. Uz to je također dodana podrška za Nintendo Switch i uređaje za virtualnu stvarnost bez koje ovaj projekt ne bi bilo moguće napraviti jer je rađen u virtualnom okruženju [19, 20].

Verzija Unity-a korištena u ovom radu je Unity 2020.3.14f1.



Slika 9: Unity logo

### 3.2 Valve Index

Za izradu i testiranje igre u ovom radu korišten je Valve Index uređaj. Sastoji se od jednog headseta, dva Index „Knuckles“ kontrolera i dva eksterna senzora za praćenje pokreta u realnom vremenu. Index koristi IPS (*eng. in-plane switching*) tehnologiju s LCD panelima od 1440x1600 piksela za svako oko što sveukupno dovodi do rezolucije od 2880x1600 piksela. Uređaj može raditi na frekvencijama osvježavanja od 80 Hz, 90 Hz, 120 Hz i eksperimentalnih 144 Hz s vidnim poljem od 120 stupnjeva. Svaki od kontrolera ima upravljačku palicu, podlogu osjetljivu na dodir, dvije glavne tipke, gumb za izbornik i okidač. Također sadrži 87 različitih senzora koji su raspoređeni po kontroleru za praćenje položaja šake i prstiju. Zvuk se reproducira putem slušalica na headsetu koje su odmaknute od ušiju korisnika [13].

Cijena kompletnog paketa s headsetom, kontrolerima i eksternim sensorima je 999 dolara te se može službeno naručiti putem platforme Steam [22].

## 4. Opis igre

Igra je pravljena u Unity engineu pomoću asseta (*hrv. dodataka*) koji su preuzeti s Asset stora unutar Unitya. Rađena je za jednog igrača i prikaz glavnog lika je u prvom licu. Postoji samo jedan level u igri koji je osmišljen kao šumovit teren s različitim objektima kao što su kuće, ograde i razni poligoni za testiranje kretanja igrača. Glavni cilj igre ne postoji, naglasak je na zabavi te isprobavanju mehanike kretanja i manipulaciji objekata. Glavni lik na raspolaganju ima par različitih vatrenih oružja te nekoliko hladnih oružja. Izgled lika je jednostavan, sastoji se samo od modeliranih ruku koje realno prikazuju korisnikove ruke u igri. Ako lik pogine, igra se ponovno pokreće i počinje ispočetka.

### 4.1 Korišteni asseti

Asset je svaki predmet koji se može koristiti za stvaranje scene ili igre [23]. Mogu se izraditi ili preuzeti unutar Unity enginea na Asset storeu, ili uvesti iz nekog drugog programa, npr. Blender. Unutar Unity-a postoji Asset Store preko kojeg se mogu kupovati i automatski ubacivati asseti u trenutni projekt. Asseti mogu biti sve od 3D modela, zvučnih zapisa, slika i drugo. U sljedećem poglavlju opisani su asseti koji su korišteni u ovom projektu.

#### 4.1.1 VRIF

VRIF (*eng. Virtual Reality Interaction Framework*) je zbirka skripti i prefabova koji služe za pomoć pri razvoju interakcija u VR igrama. Namijenjena je za game developere za lakše stvaranje vlastitih interaktivnih VR scena i bržu produktivnost. Sadrži mnogo prefabova koji se mogu odmah koristiti za stvaranje testne scene. Oni se kreću od jednostavnih blokova i lopti do prekidača, poluga te vatrorenog i hladnog oružja. Također je moguće iskoristiti vlastite modele oružja te pomoću integriranih skripti napraviti novo oružje koje će se koristiti u sceni. Unityev Prefab sustav omogućuje stvaranje, konfiguraciju i pohranu Game Objecta zajedno sa svim svojim komponentama, vrijednostima svojstava i podređenim Game Objectsima kao materijal za višekratnu upotrebu. Prefab djeluje kao predložak iz kojeg se mogu stvoriti novi objekti na sceni. Pri ponovnoj upotrebi nekog Game Objecta koji je konfiguriran na određeni način, npr. poput AI neprijatelja, rekvizita ili komada krajolika koji se pojavljuju na više mjesta u sceni ili na više scena u projektu, preporučljivo je pretvoriti ga u Prefab. Taj način je bolji od jednostavnog kopiranja i lijepljenja Game Objecta jer Prefab sustav omogućuje automatsko usklađivanje svih kopija unutar projekta.

VRIF se može pronaći na Unity-evom Asset store-u. Potrebno je kupiti asset, no moguće je probati demo verziju asseta s gotovom scenom [24].

## 4.1.2 Gaia 2 – Terrain Generator

Gaia je snažan, „all in one“ sustav za generiranje terena i scena za mobilne uređaje, VR, konzole i stolna računala. Pruža jednostavno, brzo i lijepo oblikovanje terena, teksturiranje, oblikovanje i proceduralno postavljanje sadržaja pomoću gotovih ili vlastitih uzoraka materijala. Uz malo truda i vremena, moguće je stvoriti scenu jednako kvalitetnu kao u trenutnim AAA naslovima, Slika 10.

Gaia radi na način tako da koristi različite „stamps“ (hrv. *uzorke*) terena koji se postavljaju u Unity editoru. U ovom projektu korišteni su uzorci ravnica, brda te riječnih uvala za izradu reljefa levela. Također su korišteni objekti kuća za stvaranje manjih sela, a za ukrašavanje terena bili su potrebni objekti drveća i kamenja radi popunjavanja praznina terena da scena ne izgleda dosadno. Postoje i drugi načini za izradu scena kao što su automatsko nasumično generiranje terena gdje se najbrže dobije gotova scena, ali na taj način se ne može mijenjati teren po volji korisnika. Još jedan način izrade scene je ručno izrađivanje reljefa, objekata i flore. Na taj način korisnik može mijenjati i manipulirati svakim detaljem u sceni, no time se otežava i produlji pravljenje scene. U ovom radu je odabran način rada sa „stamps“ jer uzima karakteristike oba načina koja su navedena prije. Pomoću Stampova se može brzo kreirati reljef i teren, no također se mogu postavljati flora i objekti po želji korisnika.



Slika 10: Primjer scene u Gaia generatoru

Ovaj dodatak se također može pronaći na Unityevom Asset storeu. Moguće je kupiti osnovnu verziju koja je korištena u ovom radu, no također se može kupiti i skuplja Pro verzija za profesionalnu izradu scena [25].

### **4.1.3 Emerald AI**

Emerald AI omogućuje game developerima da brzo stvore zanimljiv dinamički AI sa mnogo različitih kvalitetnih značajki kroz ili grafičko sučelje ili putem skriptiranja. Emerald Editor osmišljen je tako da stvaranje AI-a učini lakšim i prilagodljivim na mnogo različitih scenarija u igri. Emeraldov Setup Manager omogućuje korisnicima stvaranje osnovnog AI lika u vrlo kratkom roku. Emerald također uključuje 12 primjera scena, dobro dokumentirani kod i dokumentirani editor koji omogućuje nisku krivulju učenja za stvaranje AI likova. Emeraldovoj dokumentaciji, vodičima, uputama za skriptiranje i podršci pristupačno je iz uređivača.

Emerald AI se može pronaći na Asset storeu te se također može isprobati demo verzija asseta koja sadrži testnu scenu s uputama kreiranja i značajkama AI likova [26].



#### 4.1.4 Toy Soldiers

Dodatak Toy Soldiers, Slika 11, korišten je za modele AI neprijatelja. Svi modeli likova unutar paketa su plastične teksture i obojani su različitim bojama. U assetu je osam različitih modela likova gdje svaki lik predstavlja različitu klasu vojnika. Također, uz modele vojnika, postoje različiti modeli rekvizita kao što su dječje igračke, vreće s pijeskom te zastave. Za igru korišten je samo osnovni model vojnika obojan zelenom i žutom bojom s jurišnom puškom.

Dodatak Toy Soldiers se može pronaći na Asset storeu za 27 eura [27].



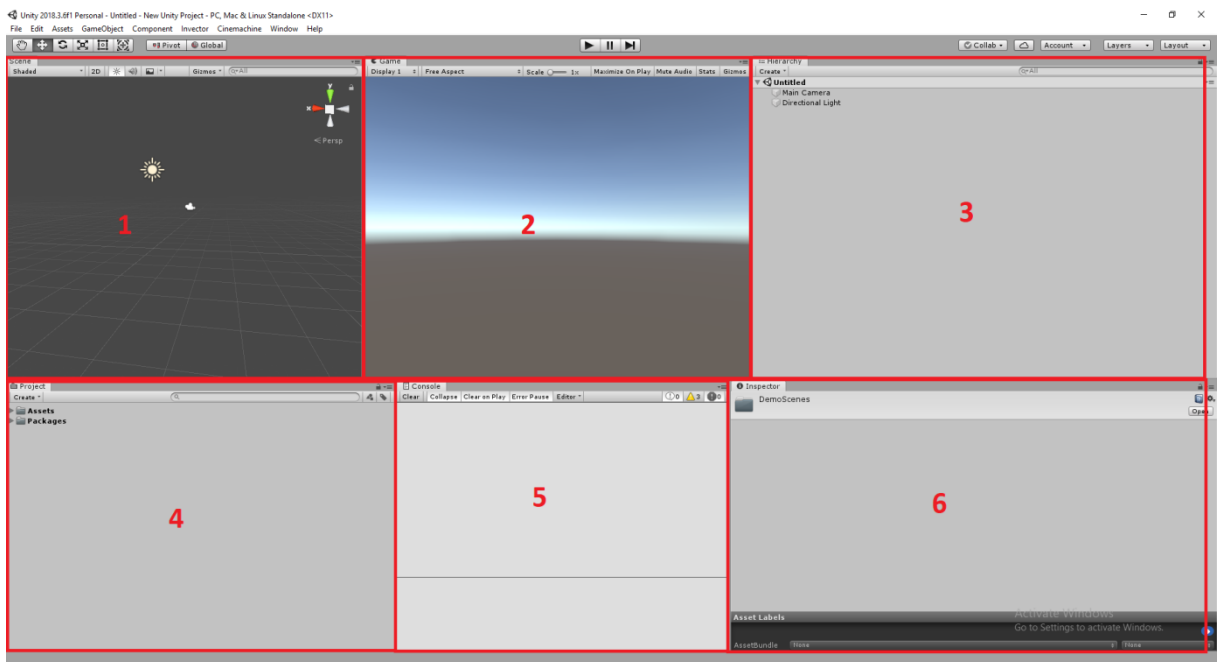
Slika 11: Modeli sadržani u dodatku



## 5. Izrada VR igre

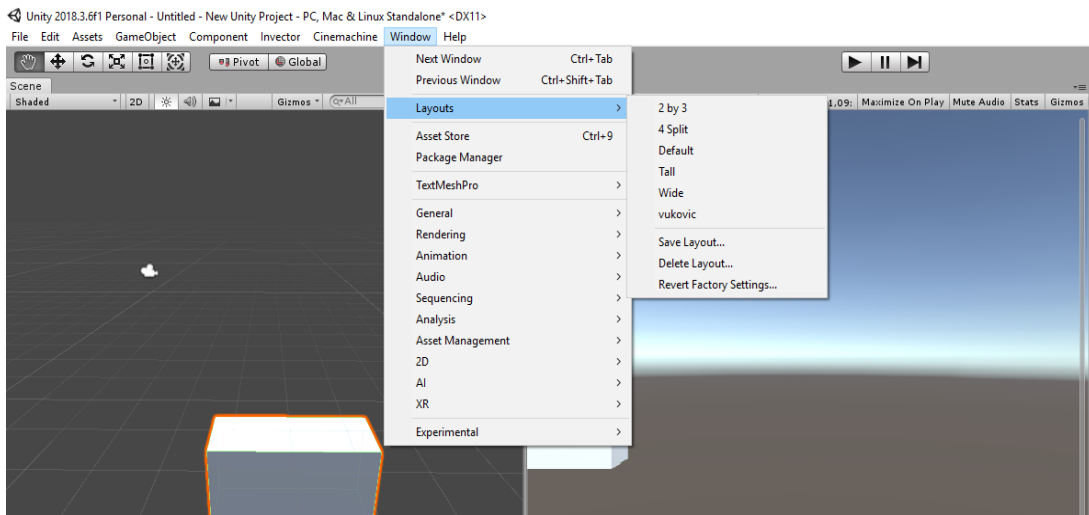
### 5.1 Podešavanje Unity Editora

Unity Editor služi za izradu i pokretanje projekata. Prije početka izrade projekta potrebno je pregledati i upoznati se s značajkama Unity Editora.



Slika 12: Raspored prozora u Unity Editor-u

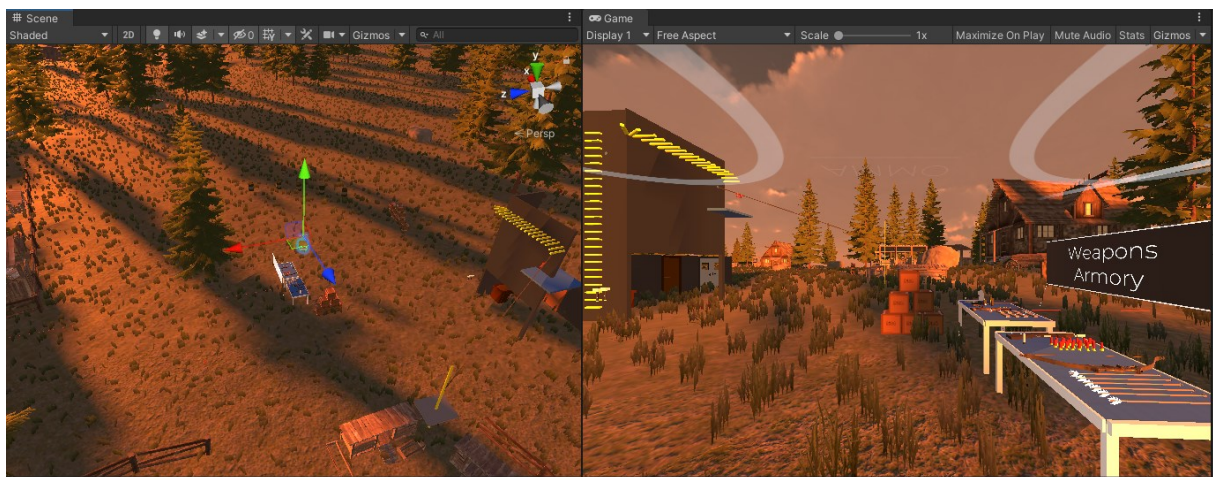
Kao što se može uočiti na slici 12, Unity Editor se može postaviti po želji korisnika. Postoji više različitih rasporeda prozora u Editoru koje je moguće isprobati pod Window -> Layouts, Slika 13. U ovom radu korišten prilagođen raspored.



**Slika 13: Različiti rasporedi za Editor**

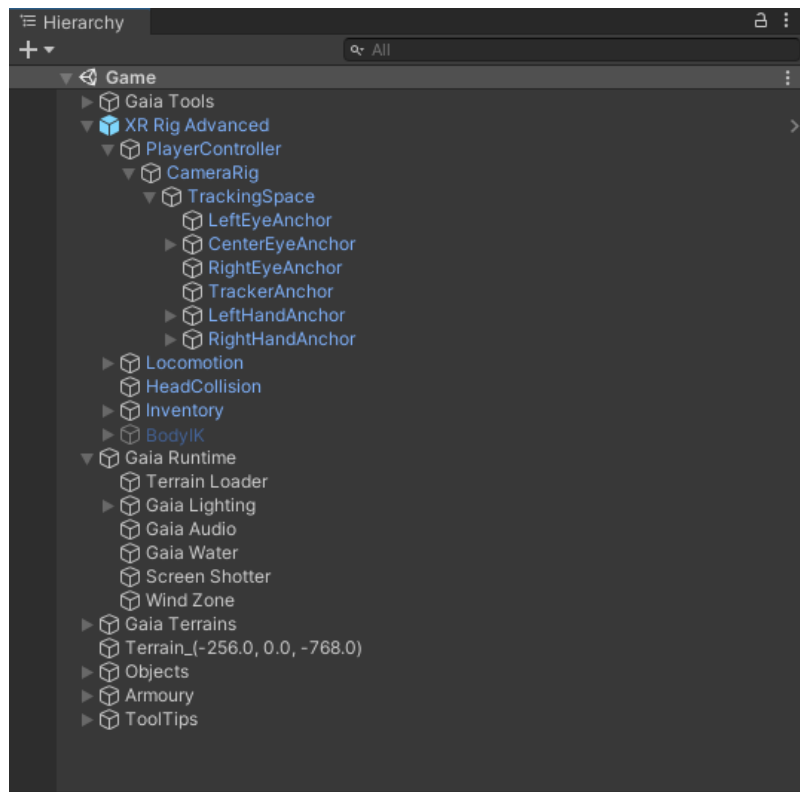
Pod brojem 1, Slika 12, nalazi se Scene prozor na kojem je moguće premještanje i uređivanje svih objekata koji se nalaze u projektu. Pomoću gumba iznad može se premješati objekte po X,Y,Z osi, povećavati ili smanjivati ih te rotirati. Također može se odrediti način pogleda na scenu da bude 3D ili 2D, ovisno o tome koja vrsta projekta se radi.

Pod brojem 2, Slika 12, nalazi se Game prozor na kojem je vidljivo kako će projekt izgledati nakon što se pokrene. Na njemu se ništa ne mijenja, nego služi samo kao pregled onoga što je napravljeno. Na tom prozoru je zapravo vidljiv pogled s glavne kamere koja je postavljena putem Scene prozora. Na slici 14 može se vidjeti razlika između Game i Scene prozora.



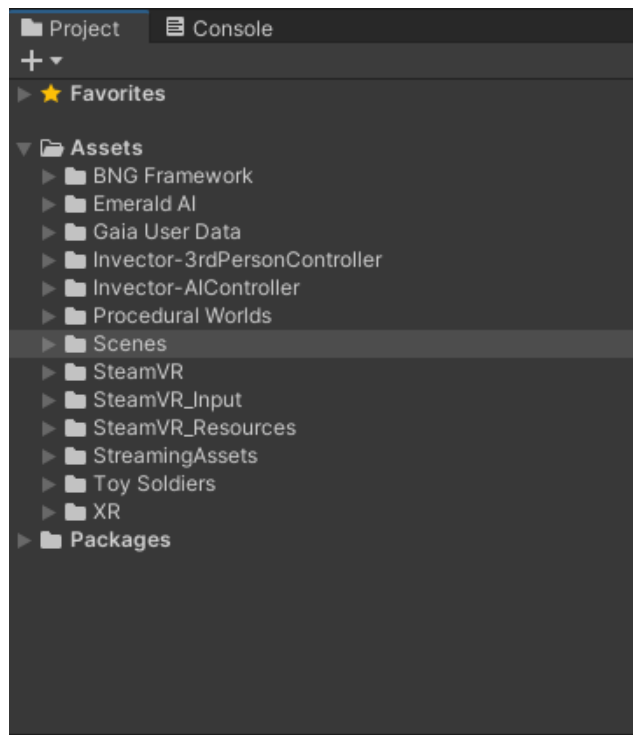
**Slika 14: Scene i Game prozor**

Pod brojem 3, Slika 12, nalazi se Hierarchy prozor na kojemu se može vidjeti hijerarhija svih objekata koji se nalaze u projektu. U tom prozoru se kreiraju instance i djeca objekata koje se koriste za stvaranje scene na Scene prozoru. Na slici 15 prikazana je hijerarhija nekoliko objekata koji su korišteni u projektu.



**Slika 15: Hijerarhija objekata**

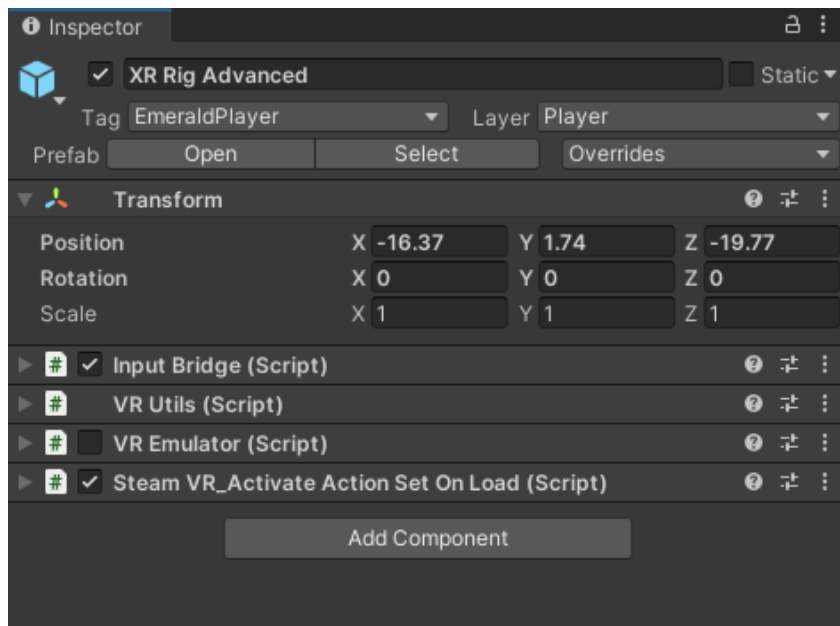
Pod brojem 4, Slika 12, nalazi se Project prozor na kojem su prikazani svi resursi, asseti, datoteke i mape koje se nalaze u projektu. Na slici 16 mogu se vidjeti korišteni resursi u ovom projektu.



**Slika 16: Korišteni resursi u projektu**

Pod brojem 5, Slika 12, nalazi se Console prozor koji služi za javljanje grešaka tijekom pokretanja scene, ako postoje. Žutim uskličnikom su prikazane greške koje nisu ozbiljne, a crvenim uskličnikom su prikazane one greške koje se moraju ispraviti da bi se scena mogla pokrenuti.

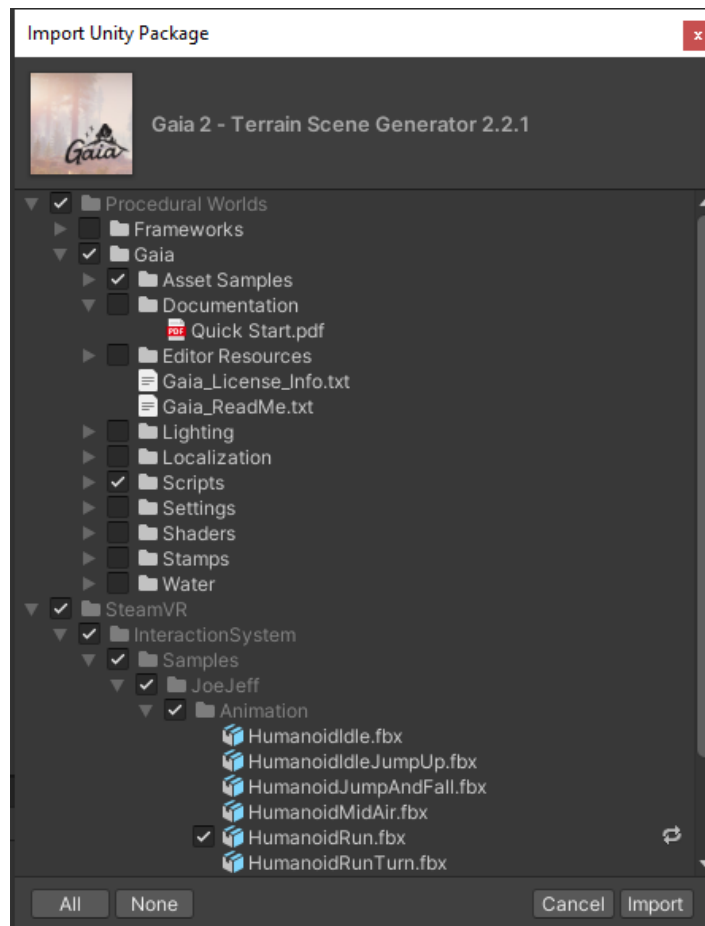
Pod brojem 6, Slika 12, nalazi se Inspector prozor. On se koristi za pregled svih postavki i svojstava pojedinog objekta. Pomoću tog prozora može se mijenjati oblik, boja i veličina, mogu se postaviti različite skripte na objekt i još mnogo toga. Na slici 17 se mogu vidjeti svojstva i naknadno dodane skripte za objekt „XR Rig Advanced“ koji je ujedno i glavni lik u igri.



Slika 17: Svojstva glavnog lika

## 5.2 Kreiranje levela

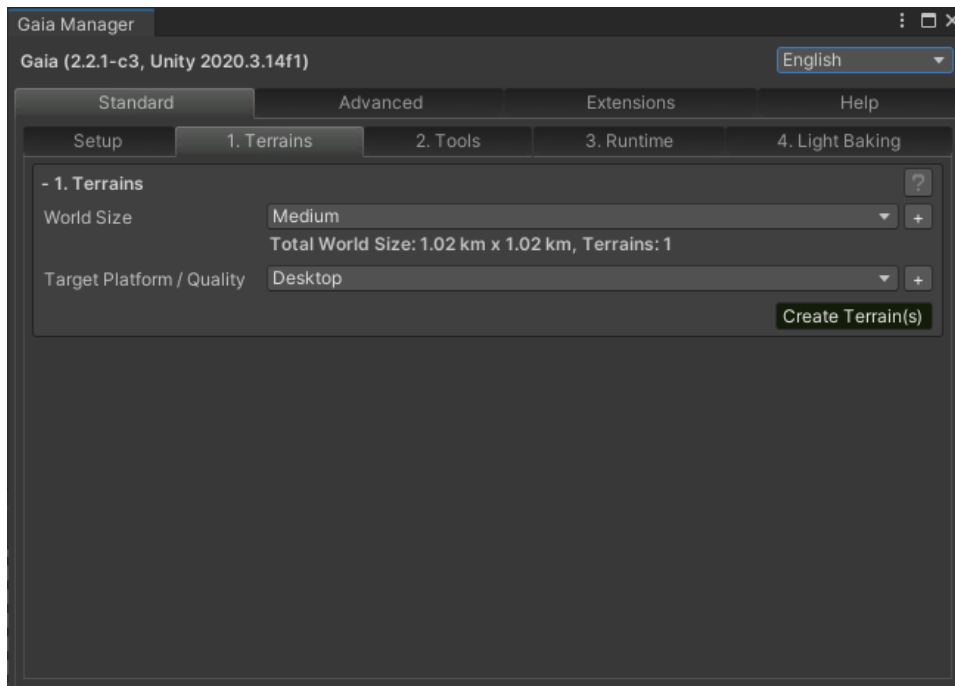
Za stvaranje levela korišten je asset Gaia 2 – Terrain Generator. Također je moguće napraviti level bez ovog asseta, što znači da bi se teren i svaki objekt u sceni pravio ručno. To bi zahtijevalo mnogo truda i vremena za izradu jednog levela. Zbog toga se koristi asset Gaia jer sadrži Prefabove objekata flore te uzorke gotovih reljefa. Prvi korak pri izradi levela je uvoz asseta u projekt. To se radi tako da se pod tabom „Assets“ klikne na „Import New Asset“ te se nakon toga odabere lokacija dodatka na vlastitom računalu. Prije uvoza će se prikazati prozor, Slika 18, na kojem su prikazane sve datoteke koje će se uvesti klikom na gumb Import.



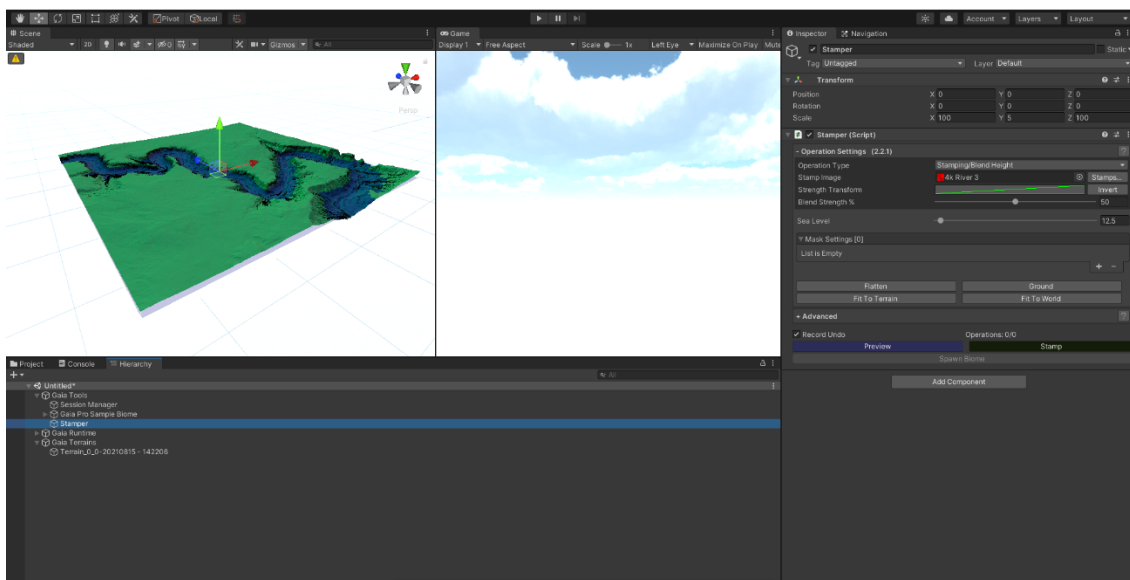
Slika 18: Gaia 2 uvoz

Nakon uvoza klikne se na tab Window -> Procedural Worlds -> Show Gaia Manager. Na ekranu će se pojaviti glavni meni Gaia-e, Slika 19, s tabovima „Setup“, „Terrains“, „Tools“, „Runtime“ i „Light Baking“. Prvo na Terrains tabu odabiremo veličinu i kvalitetu terena kojeg želimo napraviti. Za projekt je odabrana „Small“ veličina terena s dimenzijama 512\*512 metara te „Desktop“ razina kvalitete. Na sljedećem tabu Tools koristimo „Stamper“ (*hrv. nabijač*) uz pomoć kojega će se izmijeniti teren na način koji odgovara korisniku. Stamper je vrlo koristan element Gaia-e jer koristi nasumične realne uzorke terena koji mogu biti različitih oblika [28]. Stamper objekt postaviti će se u Hierarchy prozoru kao podobjekt „Gaia Tools“ objekta što znači da je spreman za korištenje. Također, pod Gaia objektom nalazi se objekt Session Manager preko kojeg se mogu vidjeti svi poduzeti koraci s oblikovanjem terena. Putem Stamper objekta u Inspector prozoru mogu se odabrati različiti oblici terena, npr. od običnih ravnica ispresijecanih rijekama do velikih planinskih lanaca, Slika 20. Stamper koji je korišten za oblik terena u projektu zove se „4K River 3“ koji prikazuje ravnicu s vijugavom rijekom koja presjeca teren na sredini. Također je korišten dodatni efekt „Blend Height“ za optimiziranje visine terena koji

izravnava i normalizira nerealne rupe i vrhove na terenu. Ta opcija se može podesiti unutar Stamper objekta pod „Operation settings“.



Slika 19: Gaia Manager

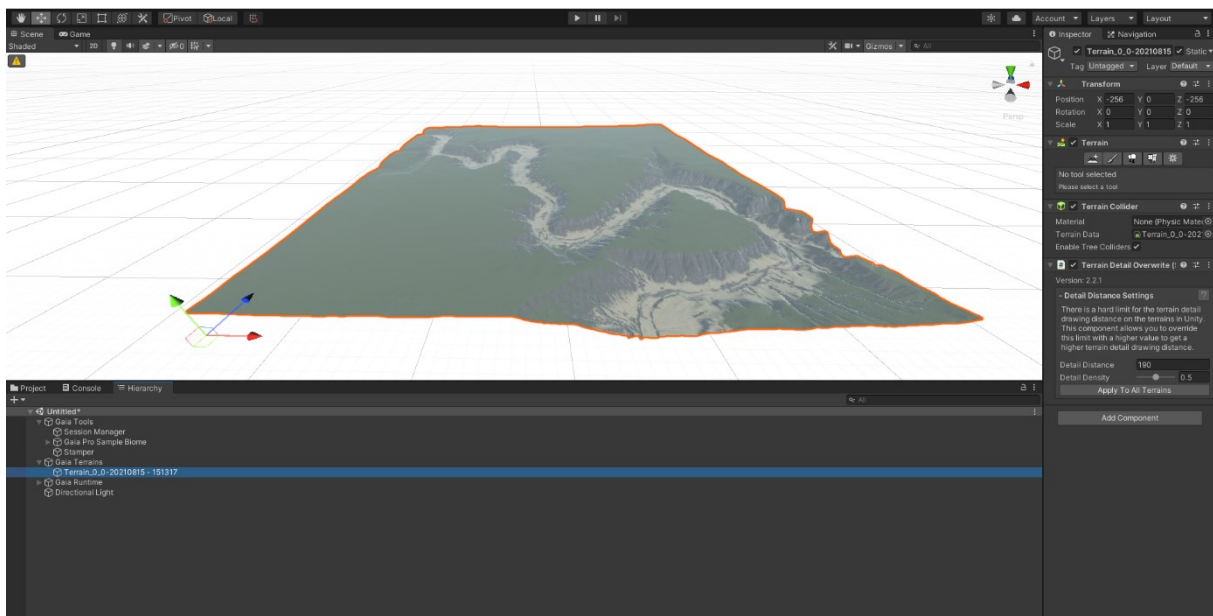


Slika 20: Postavljanje terena pomoću Stampera

Pod tabom Runtime moguće je generirati glavnog lika za testiranje terena, no u ovom projektu je zanemareno jer će se glavni lik napraviti na drugi način. Također je moguće odabrati način osvjetljenja, izgled vode i zvučke ambijenta. Voda je isključena u ovom projektu zbog bugova s korištenim VR uređajem koji iz nepoznatog razloga ne može prikazati efekt reflektiranja vode u virtualnom svijetu.

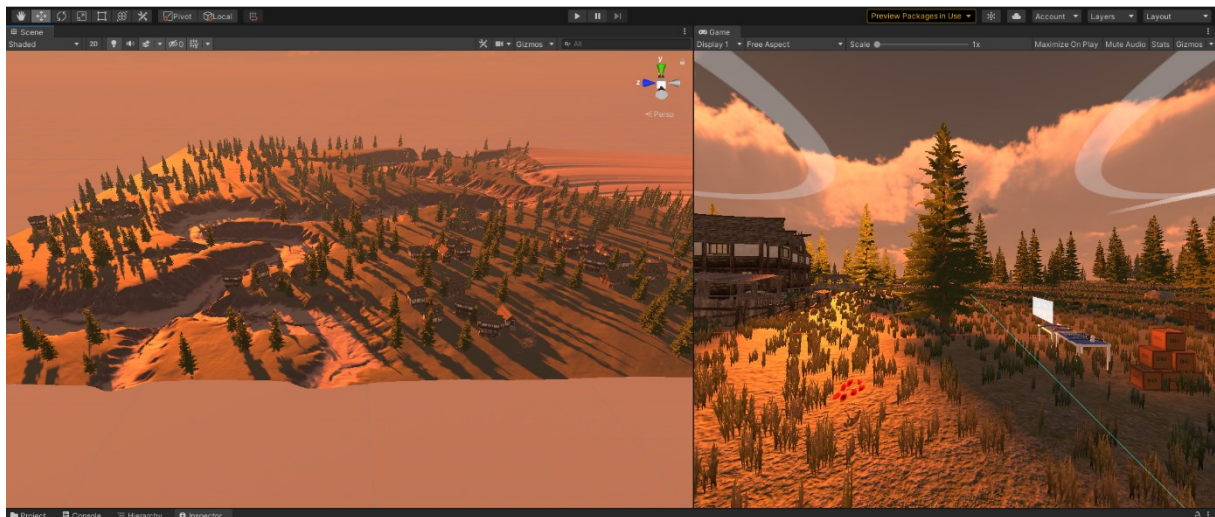
Pod tabom Light Baking potrebno je odabrati opciju „Full Lightmap Bake“ što će generirati podatke o zrakama svjetlosti koje Unity koristi za osvjetljenje scene. To su vrlo kompleksni izračuni pa je zato preporučeno to napraviti na ovaj način zbog smanjenja vremena renderiranja svjetla i sjena pri svakom pokretanju scene [29].

Nakon što se pomoću Stampera oblikuje teren, na Stamper objektu potrebno je odabrati opciju „Stamp“ što će stvoriti teren na sceni. Nakon toga, na objektu „Gaia Pro Sample Biome“, koji se nalazi pod objektom „Gaia Tools“, u Inspector prozoru pod opcijom „Biome Spawners“ odabiremo koji će se sve objekti generirati na terenu. U ovom projektu odabrani su sljedeći spawneri: Texture, Rock, Tree, Grass te Large, Medium i Small Farm Spawners. Texture Spawner će obojati teren na željeni način, Slika 21. Rock, Tree i Grass Spawner će generirati kamenje, drveće i travu po odabranoj gustoći, a Farm Spawners će stvoriti objekte kuća koje će se formirati u mala sela, Slika 22. Svaki spawner moguće je ručno podesiti da generira različite objekte, npr. pod Tree Spawnerom moguće je postaviti više vrsti drveća. Nakon završetka odabira objekata, potrebno je odabrati opciju „Spawn biome koja će generirati sve odabrane objekte po terenu. Krajnji izgled terena u projektu može se vidjeti na slici 22.



Slika 21: Level nakon dodavanja tekstura

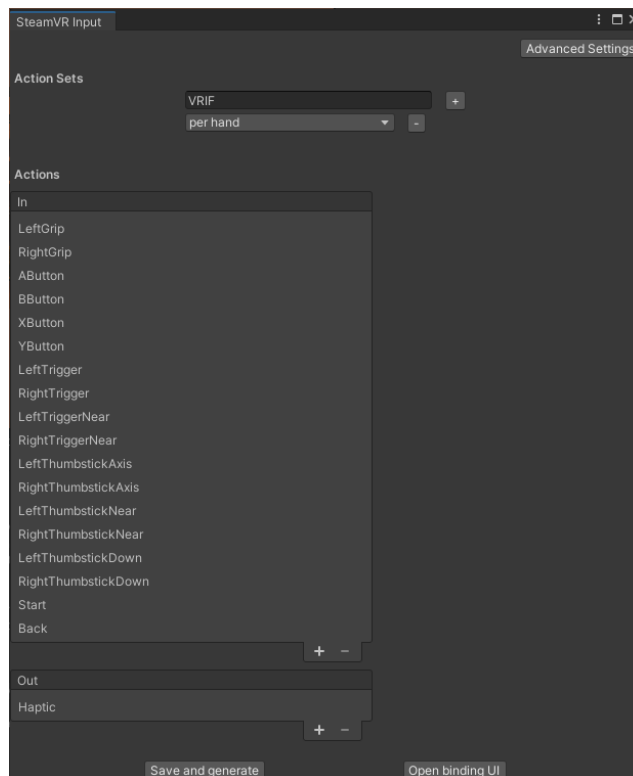




Slika 22: Krajnji izgled levela

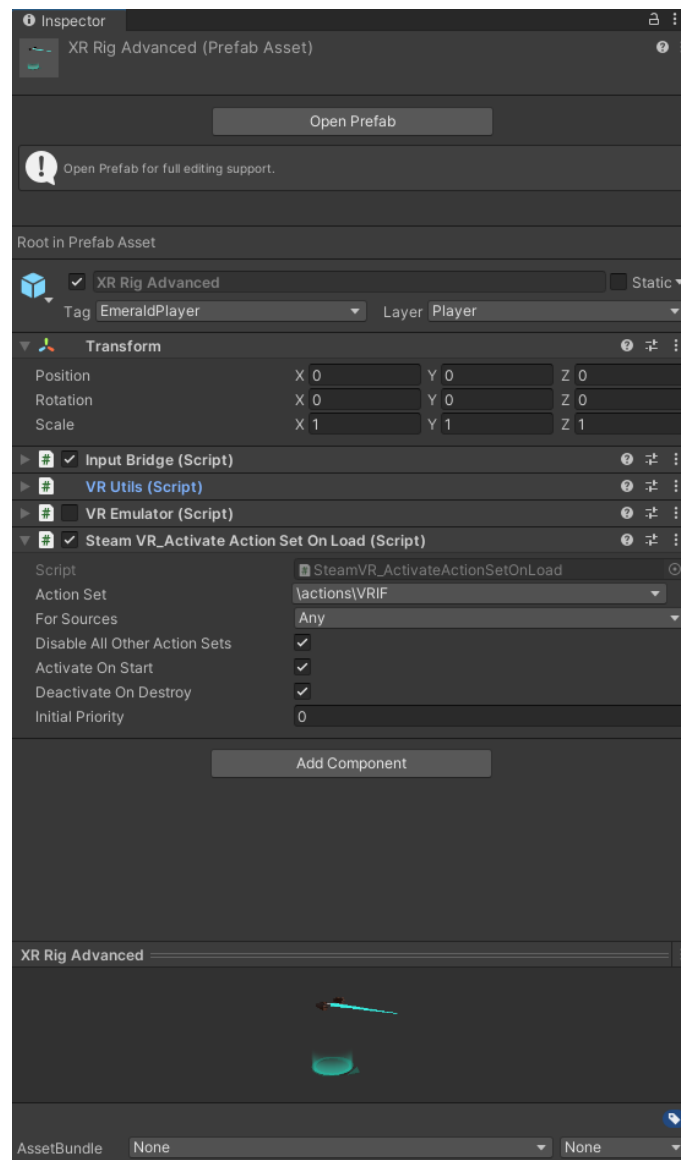
### 5.3 Kreiranje glavnog lika

Za stvaranje glavnog lika korišten je dodatak „Virtual Reality Interaction Framework“ ili skraćeno VRIF. Prvi korak je uvoz dodatka u projekt na isti način kao što je urađeno za dodatak Gaia 2 - Terrain Generator. Nakon uvoza pod Project prozorom se stvorio direktorij „BNG Framework“ koji sadrži sve asete, teksture i skripte koje će biti potrebne za izradu glavnog lika. Prije nego što se počne raditi na projektu, potrebno je prilagoditi dodatak da radi s uređajem Valve Index jer trenutno nema direktnu podršku za njega. Prvo što treba napraviti je otići na Edit -> Project Settings -> Player te pod zaglavljem Configuration postaviti opciju „Active Input Handling“ na „New“. To postavlja novi ulazni sustav koji podržava kontrole Index kontrolera. Nakon toga potrebno je postaviti shemu kontrola za Index kontrolere. VRIF dodatak ima već gotovu shemu kontrola koja se može iskoristiti. U Project prozoru potrebno je otići na BNG Framework -> Integrations -> SteamVR te odabrati asset „SteamVR“ koji će generirati shemu kontrola, Slika 23. Nakon što se generira, moguće je izmijeniti kontrole ako trenutnom korisniku ne pristaju. U ovom projektu je bilo potrebno zamijeniti kontrole za lijevi i desni upravljač jer su automatski bile krivo postavljene. Posljednja stvar koju je potrebno konfigurirati da bi Valve Index radio bez problema se nalazi pod tabom Window -> VRIF Settings. Tamo postavimo kvačicu na opciju „SteamVR Integration“ što znači da će se automatski koristiti kontrole koje smo nedavno postavili.



Slika 23: Shema kontrola

Nakon što je projekt postavljen da radi s Valve Index uređajem, može se krenuti s izradom i postavljanjem glavnog lika. U direktoriju BNG Framework -> Prefabs -> Player nalazi se objekt „XR Rig Advanced“. To je šablona glavnog lika koju će biti potrebno urediti da se može koristiti u igri. Nakon što taj objekt postavimo na scenu i kliknemo na njega, pod Inspector prozorom će se pojaviti mnogo različitih opcija, Slika 24. Pod „Input Bridge“ skriptom potrebno je postaviti opciju „Input Source“ na „Steam VR“ što znači da će objekt koristiti ulazne komande koje smo prije postavili te pod opcijom „Tracking Origin“ postavljamo „Floor“ jer želimo da je nulta točka objekta na tlu. Također je potrebno objektu postaviti Layer i Tag. Za Layer se postavlja „Player“, a za Tag je potrebno stvoriti novi koji je u projektu nazvan „EmeraldPlayer“. To će biti bitno kasnije kada će se raditi AI neprijatelji jer će putem Layera i Taga pronalaziti i ozljeđivati glavnog lika.

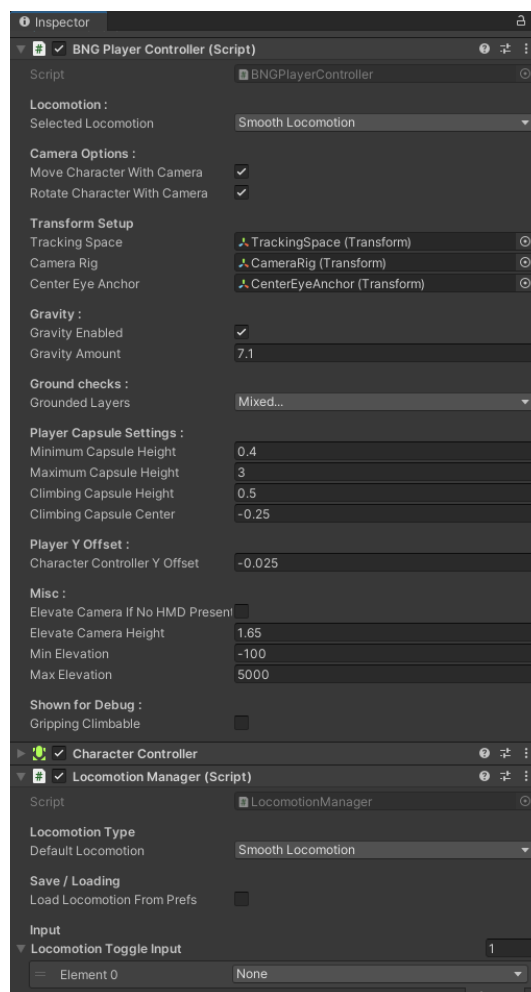


Slika 24: Opcije objekta "XR Rig Advanced"

Nakon postavljanja osnovnih opcija na „XR Rig Advanced“ objektu, potrebno je isto učiniti na njegovom podobjektu „PlayerController“ koji sadrži sve opcije vezane za kretanje, rotaciju, kameru i ponašanje glavnog lika, Slika 25 i 26.

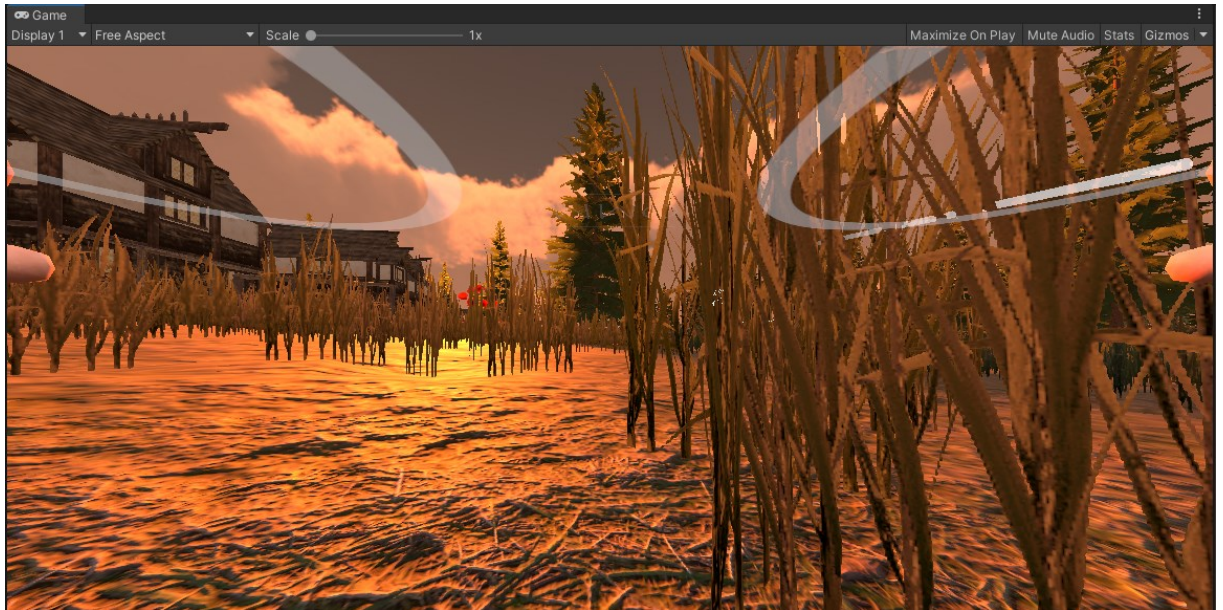
Pod opcijom „Selected Locomotion“, Slika 25, moguće je odabrati „Smooth“ što znači da će se igrač kretati konstantno ili „Teleport“ što znači da će se kretati u skokovima. Preporuča se korištenje „Teleport“ načina kretanja za one korisnike koji imaju problema s mučninom i vrtoglavicom tijekom igranja VR igara.

Postavke za kameru, Slika 25, je moguće mijenjati po potrebi. Opciju „Character Controller Y Offset“ je moguće premještati po Y osi (gore-dolje) radi prilagođavanja visine. U projektu je testirano s različitim vrijednostima te na kraju postavljeno na vrijednosti koje su prikazane na slici 25. Na slici 26 se može vidjeti početno stanje kamere gdje je vidljivo da je kamera preblizu tlu, a na slici 27 je prikazano završna pozicija kamere s podešenim Y Offsetom.

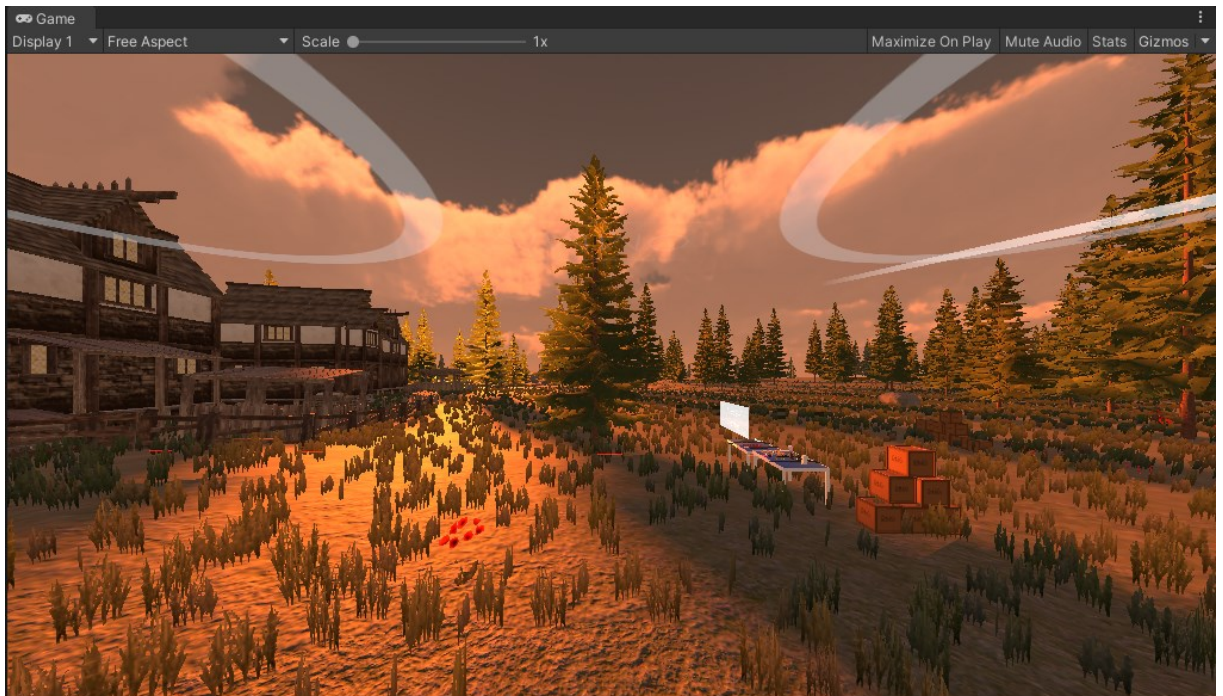


Slika 25: Postavke za kretanje i kameru





**Slika 26: Krivo postavljena visina kamere**

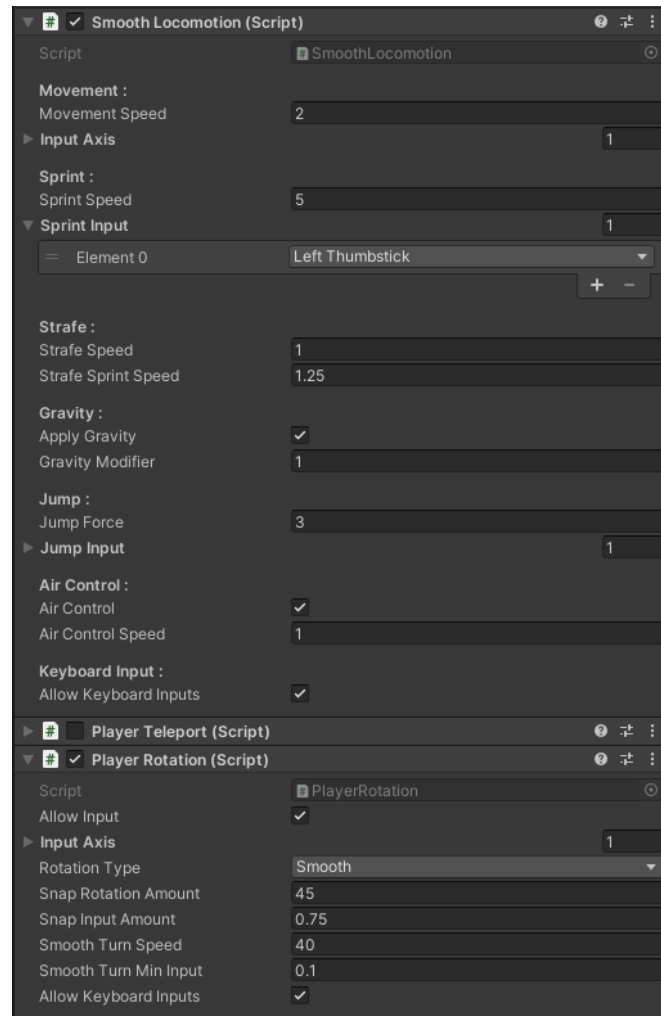


**Slika 27: Ispravljena visina kamere**

Pošto je u projektu odabran „Smooth Locomotion“ način kretanja, poželjno je odznačiti skriptu „Player Teleport“ jer neće biti potrebna. Unutar „Smooth Locomotion“ skripte moguće je podesiti brzinu hoda, trčanja i hoda u stranu. Gumb za aktivaciju trčanja postavljen je na stisak lijevog upravljača. Vrijednosti koje se koriste u projektu mogu se vidjeti na slici 28.

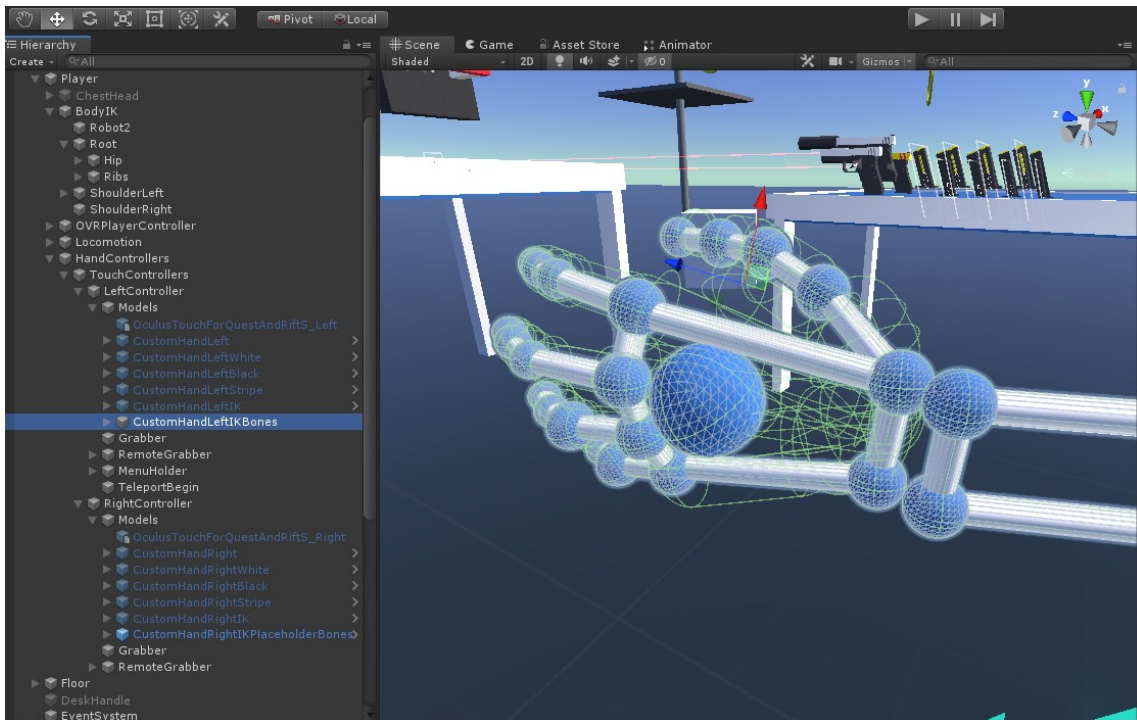
Pod postavkama rotacije moguće je odabrati „Smooth“ ili „Snap“ tip rotacije. Smooth tip rotacije radi na način tako da se igrač konstantno okreće stalnom brzinom, dok kod Snap tipa se igrač okreće za onoliko stupnjeva koliko

smo postavili u opcijama. Preporuča se korištenje Snap tipa rotacije kod korisnika koji imaju problema s vrtoglavicom i mučninom tijekom igranja VR igara. Vrijednosti postavki rotacije koje se koriste u projektu mogu se vidjeti na slici 28.



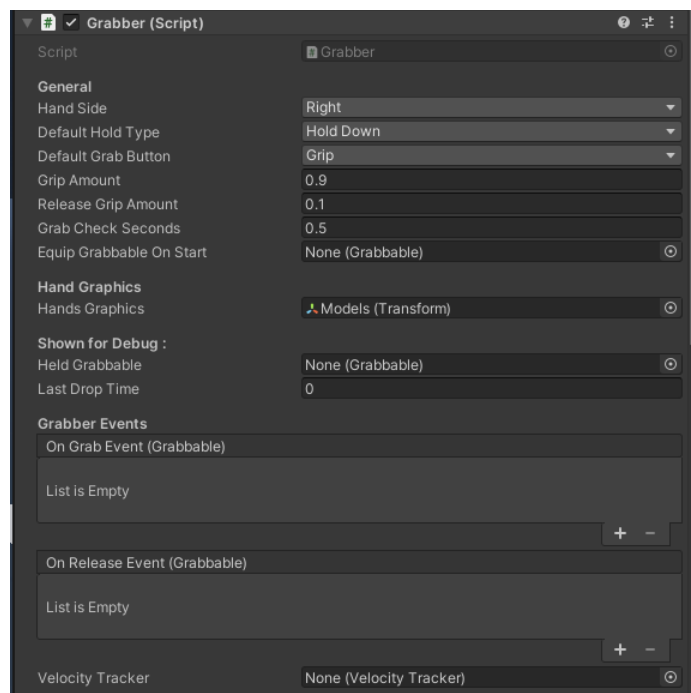
**Slika 28: Postavke za kretanje i rotaciju**

Pošto je ovo VR projekt s igranjem u prvom licu, najbitniji dio glavnog lika su ruke. Index kontroleri imaju mogućnost praćenja svakog prsta na ruci pa je zato i potrebno napraviti model ruku sa svim kostima, zglobovima i člancima da bi se ta značajka mogla u potpunosti iskoristiti. Na slici 29 je prikazan skelet te hijerarhija objekata koji su međusobno povezani da bi stvorili ruku igrača. Svaki model ruke da bi mogao funkcionirati i držati objekt mora u hijerarhiji imati podobjekt „Grabber“ koji sadrži „Trigger Collider“ koji služi za pokupljanje i podizanje ostalih objekata.



Slika 29: Skelet igrača

Grabber objekt, Slika 30, odlučuje koji objekti se mogu uhvatiti, kako će model ruke izgledati tijekom hvatanja te snaga stiska potrebna za držanje objekta. Pod postavkama je potrebno odrediti na kojoj ruci je pozicioniran Grabber (lijeva ili desna), koji tip držanja objekta će imati (stisak ili pritisak gumba) te snaga stiska. U projektu su testirane vrijednosti za pojedine opcije te su postavljene najoptimalnije vrijednosti koje se mogu vidjeti na slici 30.

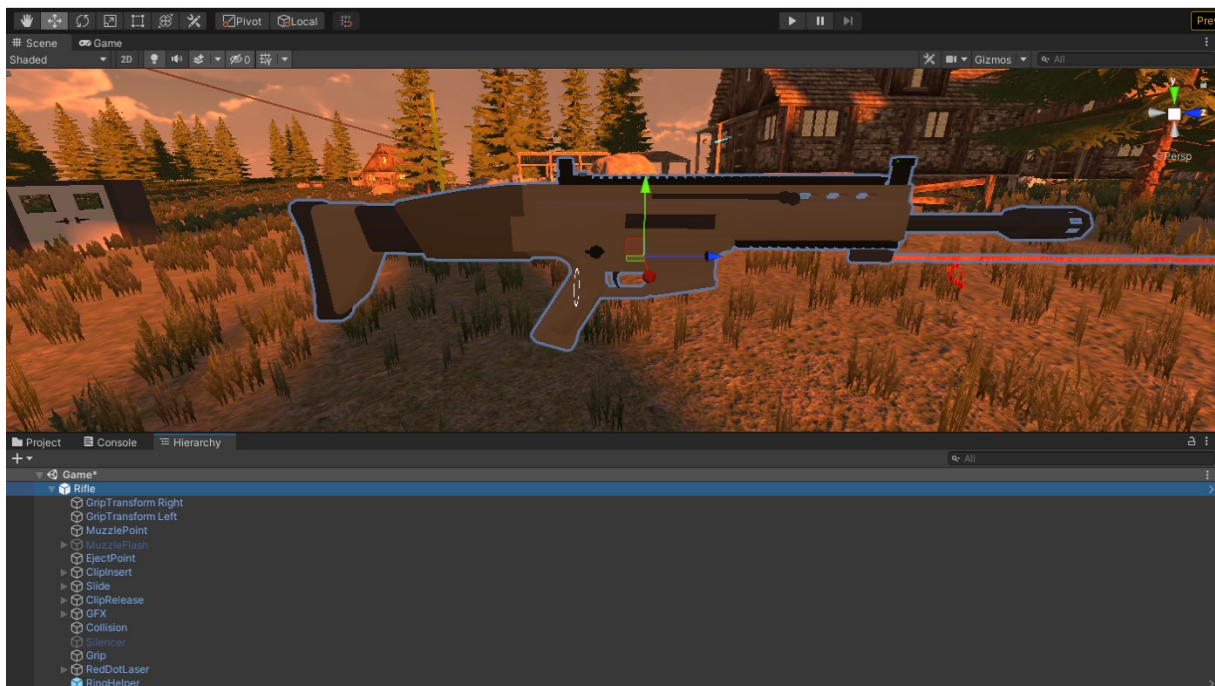


Slika 30: Postavke "Grabber" objekta



Svaki objekt koji se može primiti mora imati „Grabbable“ komponentu. Ona daje do znanja Grabber objektu kako i kada ti objekti mogu biti uzeti u ruke igrača. Također, svaki takav objekt mora imati i „Collider“ komponentu koja postavlja fizičke granice objekta i „RigidBody“ komponentu koja daje masu objektu.

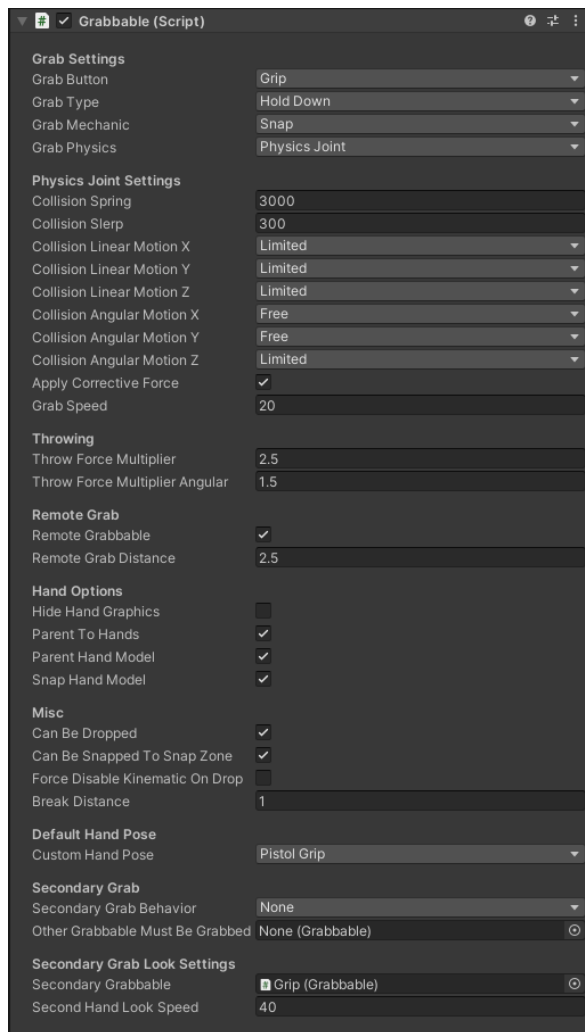
Jurišna puška, Slika 31, koja se koristi u igri rastavljena je na mnogo manjih objekata radi bolje interaktivnosti s njom. Najbitniji dijelovi su spremnik, zatvarač, okidač, rukohvat i cijev. To je potrebno jer korisnik putem gesti rukama može držati oružje na rukohvatima, izbaciti spremnik i potegnuti zatvarač što uvelike pridodaje realnosti i sveobuhvatnom iskustvu tijekom igre. Putem Grabbable komponente postavljaju se točke prihvata oružja, a zbog Collider i RigidBody komponentenata oružje će se ponašati prema ugrađenim zakonima fizike što znači da neće ostati u zraku tijekom ispuštanja iz ruku nego će padati na tlo.



Slika 31: Hijerarhija objekata oružja

Grabbable komponenta sadrži sve postavke za način hvatanja i bacanja objekta te mjesta prihvata. Pod „Grab Button“ opciji postavlja se zadani gumb za prihvata, „Grab Type“ opcija određuje da li će se prihvaćati pritiskom na gumb ili samo stiskom kontrolera, „Grab Mechanic“ određuje finoću prihvata, a „Grab Physics“ određuje kako će se objekt ponašati kad se uzme u ruke. Ostale postavke koje su bitne su vezane za bacanje objekta gdje se određuje sila izbacivanja objekta iz ruku te oblik ruku koji će se koristiti prilikom držanja oružja za točke prihvata. Na slici 32 mogu se vidjeti vrijednosti koje su korištene za Grabbable komponentu jurišne puške u projektu. Pojedine vrijednosti postavke su testirane te su postavljene one koje su najoptimalnije.





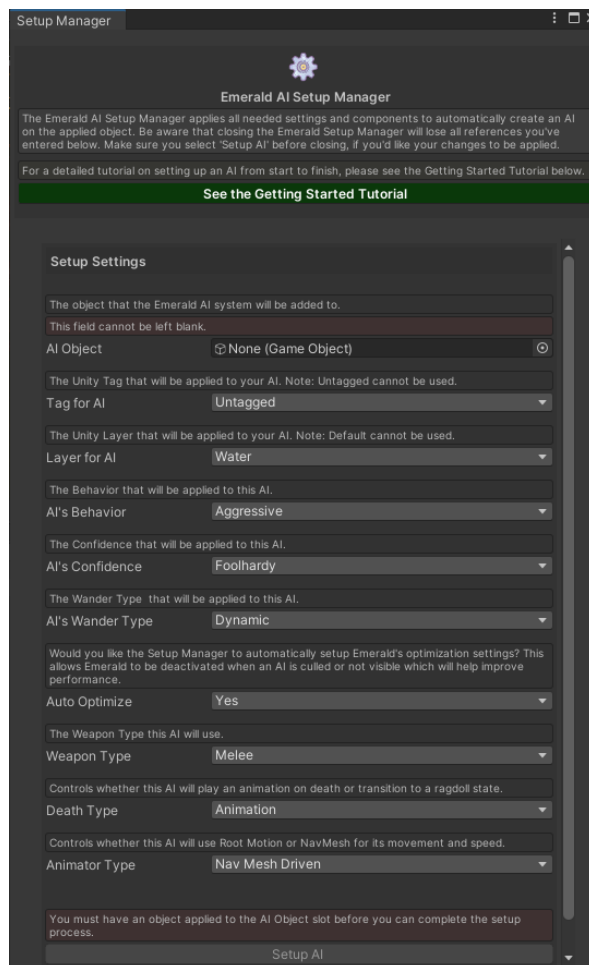
Slika 32: Postavke "Grabbable" komponente

## 5.4 Kreiranje AI neprijatelja

Za stvaranje AI neprijatelja korišten je dodatak Emerald AI verzija 2.0. Isto kao i zadnja dva dodatka, prvo je potrebno uvesti ga u projekt. Nakon uvoza se stvara tab Emerald AI pod tabom Window preko kojeg će se stvarati AI neprijatelj. Nakon što se klikne na Window -> Emerald AI -> Setup manager otvorit će se prozor za stvaranje neprijatelja, Slika 33. U manageru postoji mnogo različitih opcija, no najbitnije su:

- AI Object – služi za odabir objekta koji će se koristiti kao model neprijatelja
- Tag for AI – odabir taga (*hrv. oznaka*) koji će biti dodijeljen AI objektu
- Layer for AI – odabir layera (*hrv. sloj*) koji će biti dodijeljen AI objektu

Neke od drugih opcionalnih mogućnosti su: ponašanje, način borbe, način smrti, način hodanja i drugo. Te opcije se mogu naknadno podesiti nakon što se stvori AI objekt.



Slika 33: Setup manager

Sljedeće što je potrebno napraviti je vidno polje neprijatelja koje mu omogućuje da vidi što mu dopušta da cilja samo objekte, u ovom slučaju glavnog igrača, koje može otkriti svojim vidnim poljem. To funkcionira tako da emitira „raycast“ (*hrv. zračno polje*) prema vidnom polju odabranog kuta. Svaki objekt koji je izvan polja vidljivosti neprijatelj neće detektirati, a sve što se nalazi unutar vidnog polja će detektirati i napasti, Slika 34 i 35.

Za neprijatelja je postavljeno da se nasumično kreće po terenu. Kada uoči glavnog igrača, počinje pucati po njemu te postoji 3 slučaja završetka paljbe:

- ako je glavni igrač poginuo
- ako je glavni igrač pobjegao iz vidnog polja te neprijatelj ga nije uspio pronaći naknadno
- ako je neprijatelj poginuo

U drugom slučaju postavljeno je da neprijatelj traži glavnog igrača 10 sekundi nakon što je nestao iz vidokruga tako što se kreće prema mjestu gdje je zadnje bio viđen. Ako ga u tom intervalu pronađe, nastavlja pucati, inače se vraća nasumičnom kretanju po terenu.

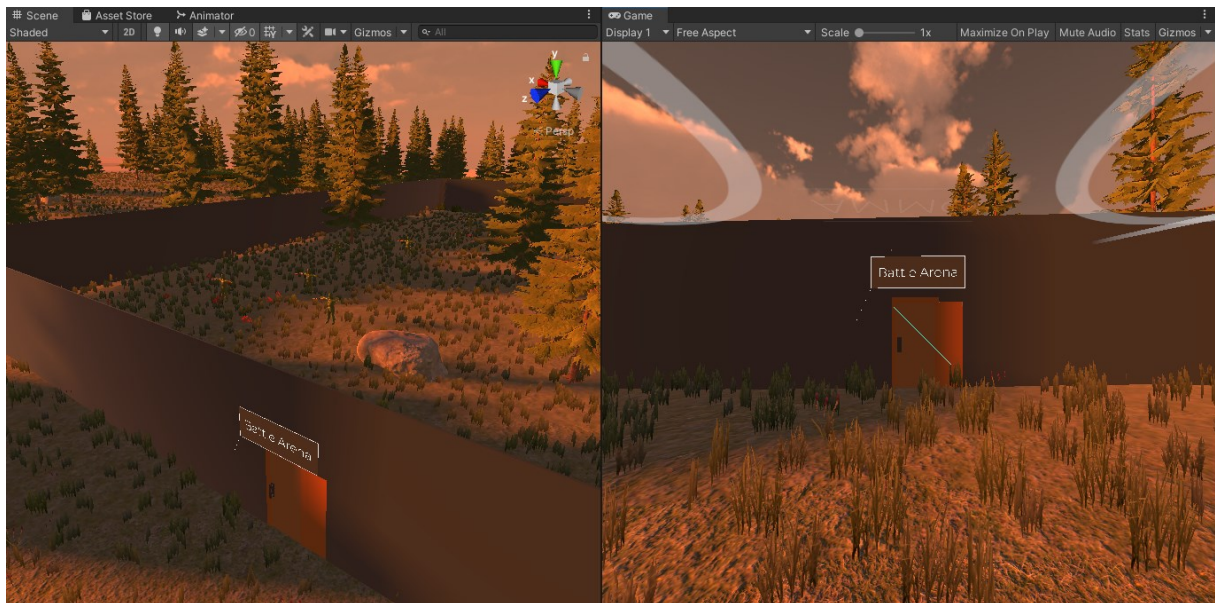


Slika 34: Igrač je u vidnom polju neprijatelja



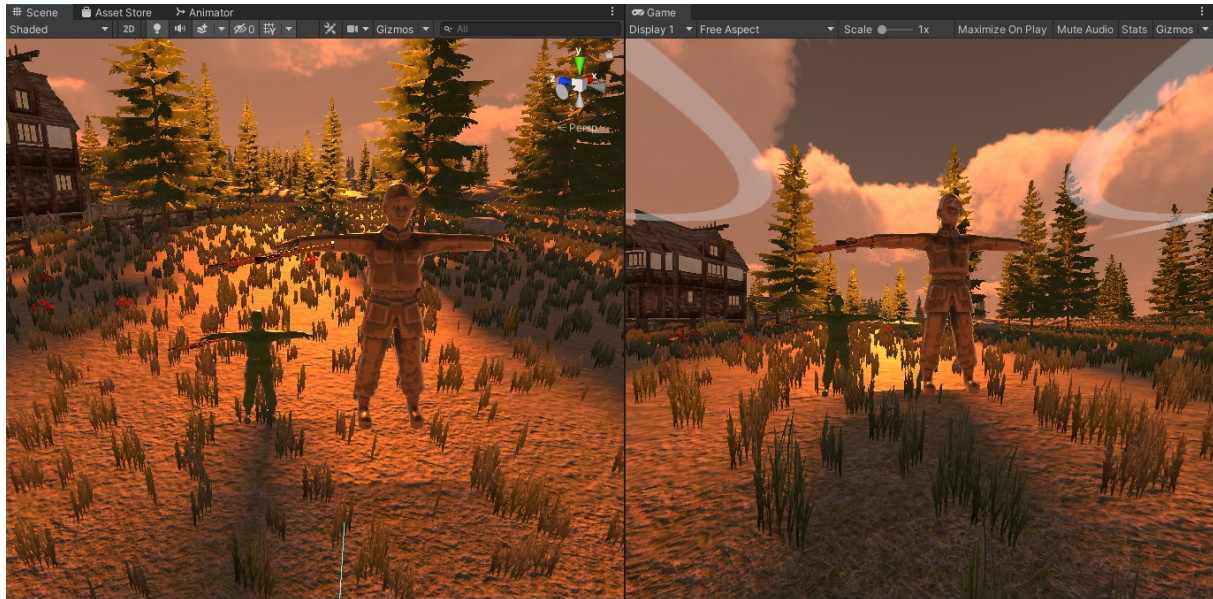
**Slika 35: Igrač je izvan vidnog polja neprijatelja**

Svi neprijatelji koji se nalaze u igri postavljeni su unutar jednog objekta, tj. borbene arene gdje se glavni igrač može boriti s njima, Slika 36. U areni je nasumičan broj neprijatelja te se razlikuju po količini zdravlja kojeg imaju te količini ozljeđivanja glavnog igrača. Jače neprijatelje je moguće prepoznati po tome što su žute boje i viši su od običnih neprijatelja, Slika 37.



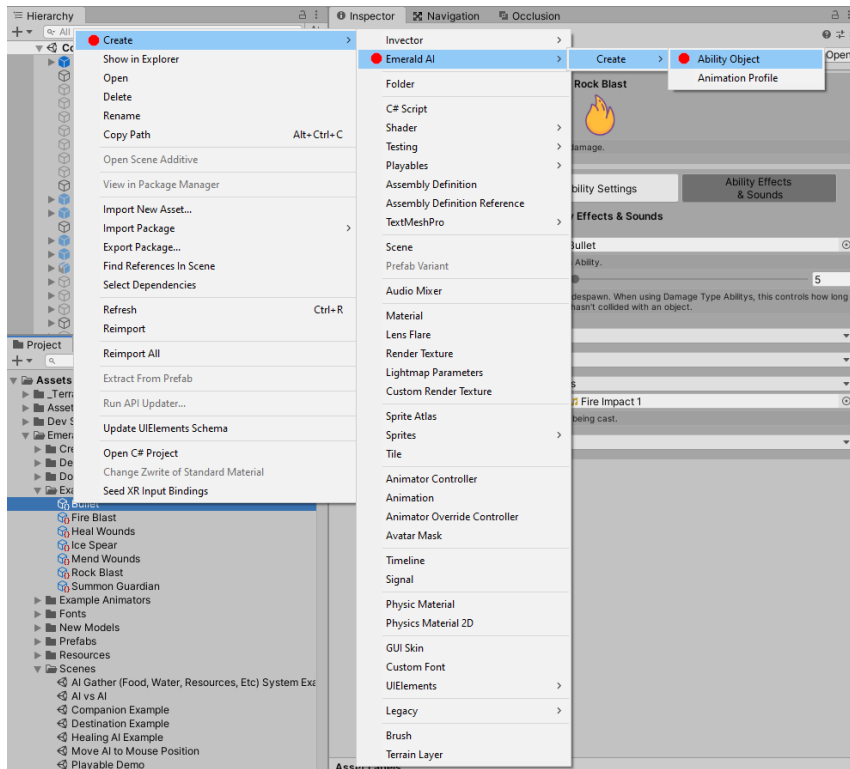
**Slika 36: Bobrena arena**



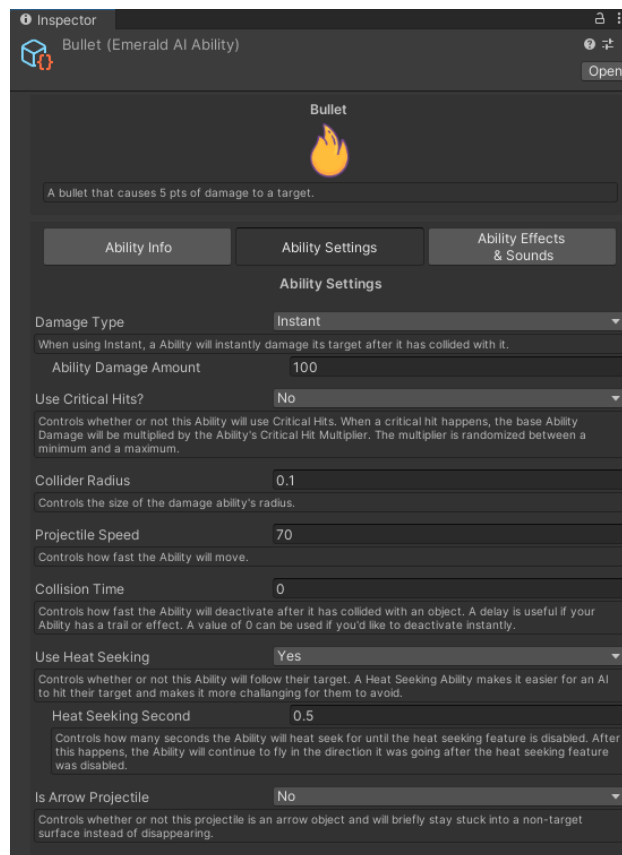


**Slika 37: Varijante neprijatelja**

Pošto ova vrsta AI neprijatelja koristi vatreno oružje, potrebno je napraviti da neprijatelj bude sposoban za dalekometnu borbu. Prvo se treba stvoriti i imenovati „Emerald AI Ability Object“. To se može napraviti tako da se stisne desni klik miša u Project prozoru i odabere se Create -> Emerald AI -> Create -> Ability Object, Slika 38. Nakon što je stvoren novi Ability Object, potrebno je prilagoditi postavke i izgled tog projektila da liči na metak. Na Inspector prozoru novonastalog projektila otvorimo „Ability Info“ tab te postavimo opciju „Ability Type“ na Damage što će omogućiti ozljeđivanje glavnog lika, Slika 39. Na „Ability Settings“ tabu se podešavaju sve postavke kolizije, kretnje i količine ozljeđivanja metka. Na slici 39 prikazane su vrijednosti koje su se koristile u projektu. Na zadnjem tabu, „Ability Effects & Sounds“, postavljen je model projektila koji liči na metak te zvukovi i efekti ispucavanja samog projektila.

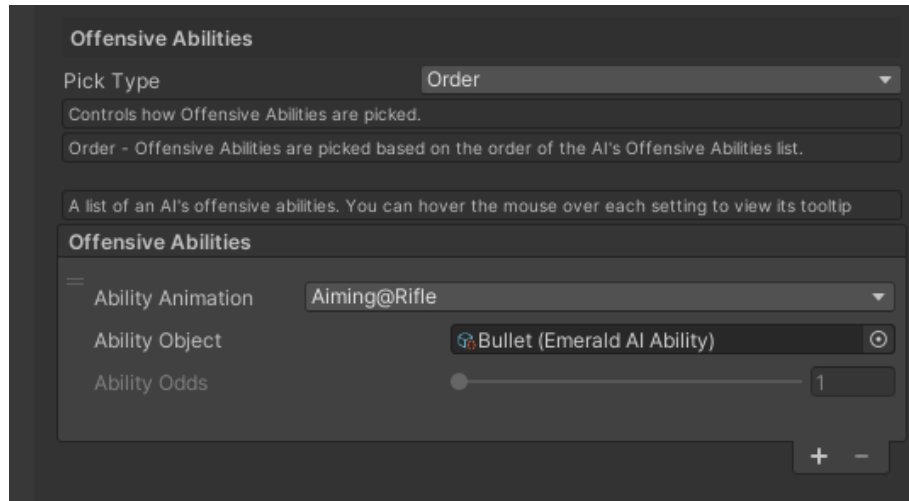


Slika 38: Stvaranje Ability Objecta



Slika 39: Vrijednosti za projektil

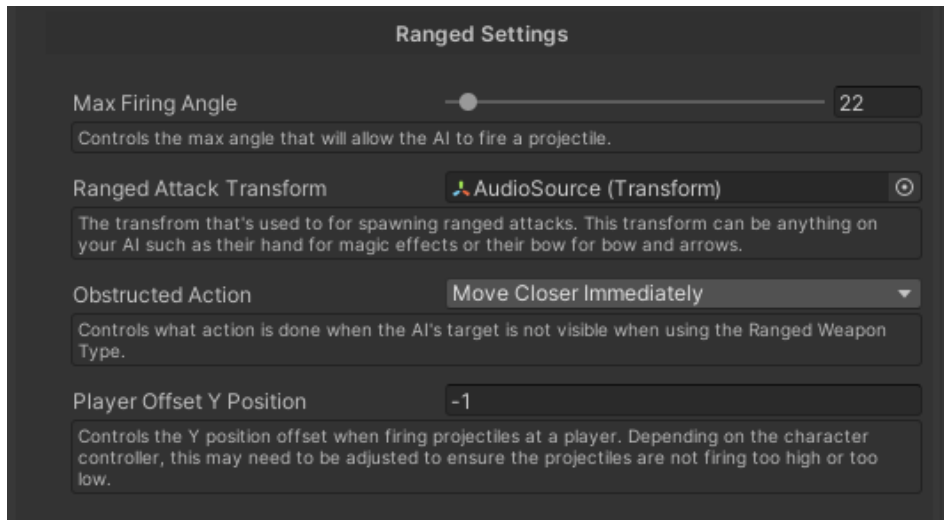
Nakon što je Ability Object stvoren i konfiguriran, potrebno ga je primijeniti na neprijatelja. Za to je potrebno otići na objekt neprijatelja te na Inspector prozoru odabrati AI's Settings -> Combat -> Damage Settings tab i otići pod „Offensive Abilities“ odjeljak. Na tom odjeljku dodajemo sposobnost koju će AI koristiti, a to je metak koji je napravljen nedavno. Uz odabir sposobnosti, također se odabire i animacija koja će se svaki puta pokretati kada AI koristi tu sposobnost, odnosno zapuca vatrenim oružjem, Slika 40.



**Slika 40: Primjena Ability Objecta na AI**

Zadnje što je potrebno napraviti da AI neprijatelj može efikasno ozlijediti glavnog igrača je postavljanje nedavno napravljenog projektila da izlazi iz cijevi vatrene oružja, a ne negdje drugdje. To se jednostavno postavi pod Inspector prozorom AI objekta kroz AI's Settings -> Combat -> Damage Settings tab. Pod odjeljkom „Ranged Settings“ postavimo „Ranged Attack Transform“ na cijev vatrene oružja što će postaviti da se projektili stvaraju na tom mjestu, Slika 41.

Također je potrebno opciju „Player Offset Y Position“ postaviti minimalno na vrijednost -1 jer inače AI neprijatelj neće detektirati glavnog igrača. Vrijednosti su testirane u projektu te je zaključeno da je to vjerojatno bug koji se javlja ako korisnik igra kao VR igrač.



**Slika 41: Postavljanje projektila da se stvara u cijevi oružja**



## 5.5 Kreiranje glavnog izbornika

Glavni izbornik je posebna scena koja se učitava nakon što se pokrene igra. Glavni izbornik se sastoji od naslova igre, gumba „Play“ i „Quit“ te slike u pozadini, Slika 42.

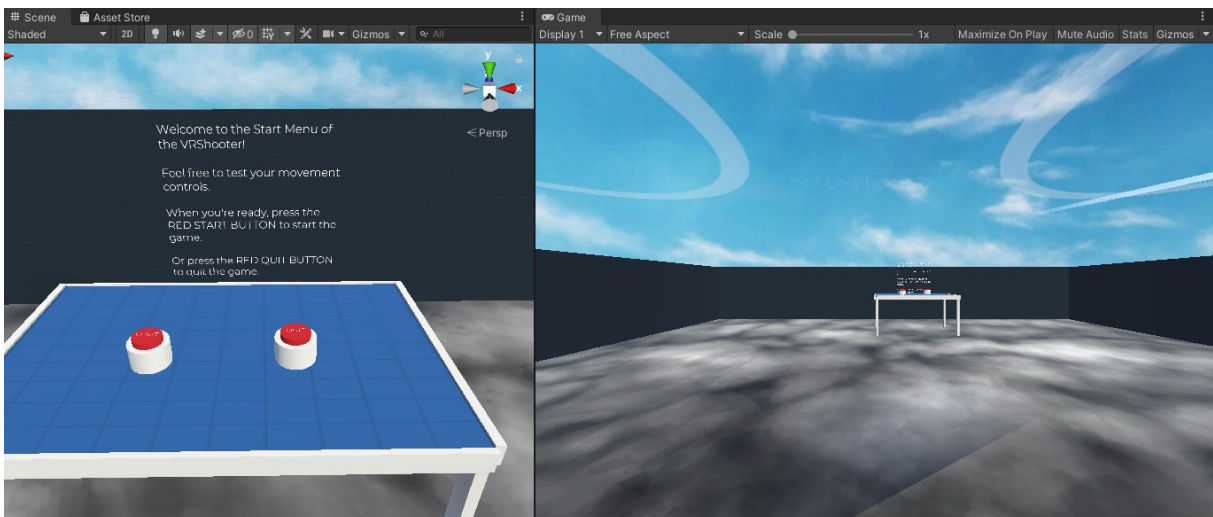
Pritiskom na gumb Play glavna scena igre se učitava, a pritiskom na Quit se izlazi iz igre. Postavljena je pozadinska slika u glavnom izborniku koja okružuje cijeli obzor radi zornog izgleda neba.

```
public class mainmenu : MonoBehaviour
{
    public void PlayGame()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

    public void QuitGame()
    {
        Debug.Log("QUIT!");
        Application.Quit();
    }
}
```

Gumb Play koristi funkciju PlayGame() koja mijenja trenutnu scenu. Indeks scene glavnog menije je 0 te se pomoću funkcije PlayGame() taj indeks poveća za 1 što pokrene ulaz u novu scenu, točnije u glavni dio igre.

Gumb Quit koristi funkciju QuitGame() koja zatvara cijelu igru te se u debug konzolu upisuje poruka „QUIT!“ zbog provjere ispravnosti.



Slika 42: Glavni izbornik

## 5.6 Kreiranje Game Over scene

Game over scena je pravljena tako da se ekran zatamni nakon što se zdravlje glavnog lika spusti na nulu. Nakon što se zatamni, prikaže se poruka da je korisnik izgubio te ga vraća na početni glavni izbornik.

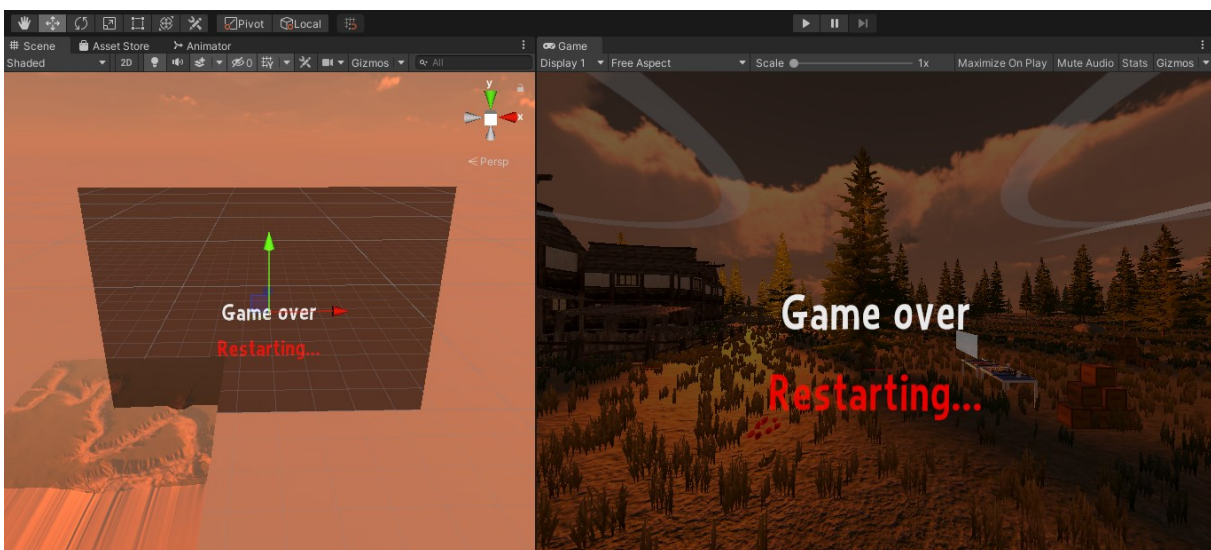
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class mainmenu : MonoBehaviour
{
    public void RestartGame(){

        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
        - 1);
    }
}
```

Funkcija RestartGame() mijenja trenutno aktivnu scenu tako da njezin indeks smanji za jedan. Pošto su samo dvije scene u igri, znamo da je indeks glavnog izbornika 0, a nivoa 1.

Game Over ekran je postavljen kao UI Canvas koji se nakon smrti glavnog lika pojavljuje preko cijelog zaslona. Unutar Unity Animatora postavljena je tranzicijska animacija gdje se lagano pojavljuje Game Over ekran kroz interval od jedne sekunde na kojemu piše „Game Over, Restarting...“, Slika 43.



Slika 43: Game Over ekran

## 6. Zaključak

Za vrijeme pisanja i izrade ovog rada i projekta sam naučio mnogo novih funkcija Unity enginea kao što je implementacija i rad s virtualnim okruženjem. Pošto već duže igram VR igre te pratim tijekom razvoja istih, bilo mi je vrlo drago raditi jedan ovakav projekt. Bilo je mnogo prepreka i teških situacija tijekom izrade, no postoji mnogo tutoriala na internetu koji su od velike pomoći. Također sam koristio pomoć i savjete drugih amaterskih developera putem jedne zajednice na Discord platformi. Završetak ove igre ne bi bio moguć bez Asset storea koji Unity pruža. Putem njega sam preuzeo mnogo raznolikih modela za objekte u igre te dodatka koji su mi bili od velike pomoći tijekom izrade projekta. Bilo je potrebno primijeniti znanje iz C# programskog jezika te iz objektno orijentiranog programiranja.

Prva verzija ove igre je predana kao završni rad na trećoj godini studija, te je sad nadograđena s VR okruženjem i predana kao diplomski rad. To ne znači da ću napustiti ovaj projekt jer imam u planu ispolirati nesavršenosti i dodati par drugih komponenti. Sljedeći korak bi bio dodavanje nivoa gađanjem glinenih goluba što bih koristio za osobni trening, a kasnije je moguće i dodavanje multiplayer komponente igri kako bi bilo moguće igrati s prijateljima.

## 7. Literatura

1. What is Virtual Reality? „<https://www.investopedia.com/terms/v/virtual-reality.asp>“ pristupljeno 9.8. 2021.
2. Virtual reality. „[https://en.wikipedia.org/wiki/Virtual\\_reality](https://en.wikipedia.org/wiki/Virtual_reality)“ pristupljeno 9.8.2021.
3. Senzorne rukavice. „<https://www.pcgamer.com/this-vr-glove-aims-to-bring-the-sense-of-touch-to-virtual-objects/>“ pristupljeno 9.8.2021.
4. Oculus Rift. [https://en.wikipedia.org/wiki/Oculus\\_Rift](https://en.wikipedia.org/wiki/Oculus_Rift) pristupljeno 9.8.2021.
5. View Master. „<https://en.wikipedia.org/wiki/View-Master>“ pristupljeno 9.8.2021.
6. Sword of Damocles. „[https://en.wikipedia.org/wiki/The\\_Sword\\_of\\_Damocles\\_\(virtual\\_reality\)](https://en.wikipedia.org/wiki/The_Sword_of_Damocles_(virtual_reality))“ pristupljeno 9.8.2021.
7. Mega Drive. „[https://en.wikipedia.org/wiki/Sega\\_Genesis](https://en.wikipedia.org/wiki/Sega_Genesis)“ pristupljeno 9.8.2021.
8. Virtuality. „[https://en.wikipedia.org/wiki/Virtuality\\_\(product\)](https://en.wikipedia.org/wiki/Virtuality_(product))“ pristupljeno 9.8.2021.
9. Virtual reality headsets. „[https://en.wikipedia.org/wiki/Virtual\\_reality\\_headset](https://en.wikipedia.org/wiki/Virtual_reality_headset)“ pristupljeno 9.8.2021.
10. Oculus Rift. „[https://en.wikipedia.org/wiki/Oculus\\_Rift](https://en.wikipedia.org/wiki/Oculus_Rift)“ pristupljeno 9.8.2021.
11. Playstation VR. „[https://en.wikipedia.org/wiki/PlayStation\\_VR](https://en.wikipedia.org/wiki/PlayStation_VR)“ pristupljeno 2.9.2019.
12. HTC Vive „[https://en.wikipedia.org/wiki/HTC\\_Vive](https://en.wikipedia.org/wiki/HTC_Vive)“ pristupljeno 9.8.2021.
13. Valve Index „[https://en.wikipedia.org/wiki/Valve\\_Index](https://en.wikipedia.org/wiki/Valve_Index)“ pristupljeno 9.8.2021.
14. Rust LTD. „<https://rustltd.com/>“ pristupljeno 9.8.2021.
15. H3VR. „[https://h3vr.fandom.com/wiki/Main\\_Page](https://h3vr.fandom.com/wiki/Main_Page)“ pristupljeno 9.8.2021.
16. Steam Awards. „<https://store.steampowered.com/sale/BestOf2020?tab=3>“ pristupljeno 9.8.2021.
17. Game engine. „[https://en.wikipedia.org/wiki/Game\\_engine](https://en.wikipedia.org/wiki/Game_engine)“ pristupljeno 9.8.2021.
18. Unreal Engine. „[https://en.wikipedia.org/wiki/Unreal\\_Engine](https://en.wikipedia.org/wiki/Unreal_Engine)“ pristupljeno 9.8.2021.
19. Unity engine. „[https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))“ pristupljeno 9.8.2021.
20. Unity. „<https://unity3d.com/unity>“ pristupljeno 9.8.2021.
21. Razlika između Unity-a i Unreal Engine-a. „<https://hackr.io/blog/unity-vs-unreal-engine>“ pristupljeno 9.8.2021.
22. Steam Valve Index purchase. „<https://store.steampowered.com/valveindex>“ pristupljeno 9.8.2021.
23. Assets <https://unity3d.com/quick-guide-to-unity-asset-store> pristupljeno 9.8.2021.
24. VRIF „<https://assetstore.unity.com/packages/templates/systems/vr-interaction-framework-161066>“ pristupljeno 10.8.2021.

25. Gaia 2 – Terrain Generator.  
„<https://assetstore.unity.com/packages/tools/terrain/gaia-2-terrain-scene-generator-42618>“ pristupljeno 10.8.2021.
26. Emerald AI. „<https://assetstore.unity.com/packages/tools/ai/emerald-ai-2-0-40199>“ pristupljeno 10.8.2021.
27. Toy Soldiers. <https://assetstore.unity.com/packages/3d/characters/toy-soldiers-characters-and-weapons-61368> pristupljeno 10.8.2021.
28. Gaia 2 Stamper.  
„<https://proceduralworlds.freshdesk.com/support/solutions/articles/33000254749-stamper-introduction>“ pristupljeno 14.8.2021.
29. Unity Baked Lights. <https://docs.unity3d.com/Manual/LightMode-Baked.html> pristupljeno 14.8.2021.

## 8. Popis slika

Slika 1: View Master .....	2
Slika 2: Damoklov mač .....	3
Slika 3: Oculus Rift .....	4
Slika 4: Playstation VR .....	4
Slika 5: HTC Vive .....	5
Slika 6: Valve Index .....	5
Slika 7: H3VR .....	6
Slika 8: Scena iz igre .....	6
Slika 9: Unity logo .....	8
Slika 10: Primjer scene u Gaia generatoru .....	10
Slika 11: Modeli sadržani u dodatku .....	12
Slika 12: Raspored prozora u Unity Editor-u .....	13
Slika 13: Različiti rasporedi za Editor .....	14
Slika 14: Scene i Game prozor .....	14
Slika 15: Hijerarhija objekata .....	15
Slika 16: Korišteni resursi u projektu .....	16
Slika 17: Svojstva glavnog lika .....	17
Slika 18: Gaia 2 uvoz .....	18
Slika 19: Gaia Manager .....	19
Slika 20: Postavljanje terena pomoću Stamera .....	19
Slika 21: Level nakon dodavanja tekstura .....	20
Slika 22: Krajnji izgled levela .....	21
Slika 23: Shema kontrola .....	22
Slika 24: Opcije objekta "XR Rig Advanced" .....	23
Slika 25: Postavke za kretanje i kameru .....	24
Slika 26: Krivo postavljena visina kamere .....	25
Slika 27: Ispravljena visina kamere .....	25
Slika 28: Postavke za kretanje i rotaciju .....	26
Slika 29: Skelet igrača .....	27
Slika 30: Postavke "Grabber" objekta .....	27
Slika 31: Hijerarhija objekata oružja .....	28
Slika 32: Postavke "Grabbable" komponente .....	29
Slika 33: Setup manager .....	30
Slika 34: Igrač je u vidnom polju neprijatelja .....	31
Slika 35: Igrač je izvan vidnog polja neprijatelja .....	32
Slika 36: Borbena arena .....	32
Slika 37: Varijante neprijatelja .....	33
Slika 38: Stvaranje Ability Objecta .....	34
Slika 39: Vrijednosti za projektil .....	34
Slika 40: Primjena Ability Objecta na AI .....	35
Slika 41: Postavljanje projektila da se stvara u cijevi oružja .....	36

Slika 42: Glavni izbornik .....	37
Slika 43: Game Over ekran .....	38



## **9. Prilozi**

Uz rad priložena je VR igra sa svim korištenim komponentama.