

Alan Thorn

Naučíte Unity 5.x

Kreirajte izvanredne igre na profesionalnom nivou.

Alan Thorn

Naučite **Unity 5.x**

Kreirajte izvanredne igre na profesionalnom nivou.



 **kompjuter
biblioteka**



Izdavač:



Obalskih radnika 15, Beograd

Tel: 011/2520272

e-mail: kombib@gmail.com

internet: www.kombib.rs

Urednik: Mihailo J. Šolajić

Za izdavača, direktor:

Mihailo J. Šolajić

Autor: Alan Thorn

Prevod: Slavica Prudkov

Lektura: Miloš Jevtović

Slog : Zvonko Aleksić

Znak Kompjuter biblioteke:

Miloš Milosavljević

Štampa: „Pekograf“, Zemun

Tiraž: 500

Godina izdanja: 2017.

Broj knjige: 492

Izdanje: Prvo

ISBN: 978-86-7310-515-4

Mastering Unity 5.x

by Alan Thorn

ISBN 978-1-78588-074-2

Copyright © 2017 Packt Publishing

All right reserved. No part of this book may be reproduced or transmitted in any form or by means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher. Autorizovani prevod sa engleskog jezika edicije u izdanju „Packt Publishing“, Copyright © 2017.

Sva prava zadržana. Nije dozvoljeno da nijedan deo ove knjige bude reprodukovan ili snimljen na bilo koji način ili bilo kojim sredstvom, elektronskim ili mehaničkim, uključujući fotokopiranje, snimanje ili drugi sistem presnimavanja informacija, bez dozvole izdavača.

Zaštitni znaci

Kompjuter Biblioteka i „Packt Publishing“ su pokušali da u ovoj knjizi razgraniče sve zaštitne oznake od opisnih termina, prateći stil isticanja oznaka velikim slovima.

Autor i izdavač su učinili velike napore u pripremi ove knjige, čiji je sadržaj zasnovan na poslednjem (dostupnom) izdanju softvera. Delovi rukopisa su možda zasnovani na predizdanju softvera dobijenog od strane proizvođača. Autor i izdavač ne daju nikakve garancije u pogledu kompletnosti ili tačnosti navoda iz ove knjige, niti prihvataju ikakvu odgovornost za performanse ili gubitke, odnosno oštećenja nastala kao direktna ili indirektna posledica korišćenja informacija iz ove knjige.

O AUTORU

Alan Thorn je multidisciplinarni programer igara, autor i predavač (sa iskustvom od 17 godina). On kreira igre za PC desktop, mobilne uređaje i VR. Osnovao je Wax Lyrical Games i kreirao je nagrađivanu igru Baron Wittard: Nemesis of Ragnarok, radeći na njoj kao dizajner, programer i umetnik. Napisao je 23 tehničke knjige o razvoju igara i predstavio je 19 video obuka koje obuhvataju programiranje igara, programiranje u Unityju, 3D modelovanje i animacije. Podučavao je studente o kreiranju igara kao viši predavač na Univerzitetu Teesside, kao vodeći predavač za Uppingham School i kao gostujući predavač za London South Bank University. Trenutno je na čelu odeljenja za Games Design and Development u školi National Film and Television School.

PRIZNANJA

Ova knjiga ne bi postojala da nije bilo velike pomoći mnogih vrednih ljudi. Zahvaljujem se svim mojim prijateljima, mojoj porodici i timu u „Packt“ izdavačkoj kući.

O RECENZENTU

Francesco Sapio stekao je svoju diplomu iz računarske nauke i inženjeringa na University of Rome, u Italiji, nekoliko semestara pre roka sa najvišom ocenom. Trenutno priprema magistraturu inženjeringa u veštačkoj inteligenciji i robotici na istom univerzitetu.

On je Unity 3D i Unreal stručnjak, kvalifikovani programer igara i iskusni korisnik osnovnih grafičkih programa. Kreirao je edukativnu igru za učenike srednjih škola za učenje nekih koncepata iz fizike Game@School (Sapienza University of Rome) i seriju igara za decu Sticker Book (Dataware Games). Osim toga, radio je kao konsultant za igru Prosperity – Italy 1434 (Entertainment Game Apps, Inc), koju je uspešno osnovao Kickstarter, i za otvoreni online zajednički idejni sistem pod naslovom Inno-voice (Sapienza University of Rome). Osim toga, bio je uključen u različite istraživačke projekte, kao što je Belief Driven Pathfinding (Sapienza University of Rome), koji je nova tehnika za pathfinding u video igrama predstavljenim kao novina na DiGRAFDG Conference 2016 i perfekt.ID (Royal Melbourne Institute of Technology) i koji uključuje razvoj sistema za preporuke za igre.

On je aktivni obrađivač tema o razvoju igara. Nedavno je napisao knjigu „Unity UI Cookbook“, koja podučava čitaoca kako da kreiraju uzbudljive i praktične korisničke interfejsa za igre unutar Unityja, a napisao je i kratak e-vodič „What do you need to know about Unity“. Pored toga, koautor je knjige „Unity 5.x 2D Game Development Blueprints“. Takođe je bio recenzent za knjige „Game Physics Cookbook“, „Unity 5.x by Example“ i „Unity Game Development Scripting“. Izdavač ovih pet knjiga i e-vodiča je „Packt“.

Francesco voli matematiku, filozofiju, logiku i rešavanje slagalica, ali najviše kreiranje video igara, zahvaljujući svojoj strasti za dizajniranje i programiranje igara.

Možete se povezati sa njim na adresi www.francescosapio.com.



UVOD

U ovoj knjizi ćemo najpre da opišemo Unity sa tačke gledišta početnika, a zatim ćemo vam pomoći da stečeno znanje razvijete na određeni način. Konkretno, steći ćete osnovno sveobuhvatno umeće, na osnovu kojeg ćete naučiti Unity tako da kao iskusni, nezavisni programer možete da radite u bilo kom odeljenju i pomoću bilo kojeg seta funkcija unutar Unityja. U sledećem opisu poglavlja prikazan je detaljno ceo raspon funkcija koje ćemo istražiti. Naš osnovni cilj je da vas ova knjiga učini svestranim i potpuno kompetentnim u Unityju i da sposobnim da rešite probleme sa kojima se možete susresti u setu funkcija Unityja. U ovoj knjizi ćemo se koncentrisati, uglavnom, na praktične primere; izgrađićemo FP borbenu igru kroz više poglavlja, što će na više načina testirati vašu veštinu kucanja.

ŠTA OBUHVATA OVA KNJIGA

U Poglavlju 1, „*Priprema i konfigurisanje elemenata postavke*“, govori se o projektu na koji ćemo se fokusirati u knjizi. Specifikovani su saveti i trikovi za optimalno importovanje elemenata, konfigurisanje elemenata i tokove najbolje prakse da bi projekat ostao organizovan.

U Poglavlju 2, „*Dizajn i struktura nivoa*“, opisan je početak projektovanja i izgradnje osnovnog nivoa igre unutar Unityja, pri čemu se istražuju interesantne ideje dizajniranja nivoa.

U Poglavlju 3, „*Kontrole igrača – pokret*“, istražujemo kontrole igrača, odnosno, kako unos korisnika vodi i kontroliše glavnu kameru igre.

Poglavlje 4, „*Kontrole igrača – kucanje i zdravlje*“, fokusirano je na kreiranje borbene mehanike koja uništava neprijatelje; svaki karakter koji može da bude oštećen zahteva zdravlje.

U Poglavlju 5, „*Neprijatelji i veštačka inteligencija*“, opisano je kako se postiže inteligentno i verodostojno ponašanje neprijatelja pomoću AI-ja. Upotrebom AI-ja neprijatelj će se pomerati na nivou, tražeći igrača - napašće ga kada je u njegovom dometu.

U Poglavlju 6, „*Upravljanje projektom i verzijama*“, opisani su načini optimizovanja toka razvoja igre pomoću softvera Version Control. Ovaj softver vodi evidenciju o projektu dok se on pomera kroz sve svoje iteracije.

Poglavlje 7, „*Trajni podaci – učitavanje i snimanje stanja igre*“, fokusirano je na različite metode snimanja koji su dostupni u Unityju; igračima je omogućeno da snime svoj napredak u igri i da nastave igru na mestu na kojem su prethodno prekinuli svaki put kada započinju novu sesiju igre.

U Poglavlju 8, „*Performansa, optimizacija, mobilni uređaji i drugo*“, istražena je selekcija povezanih tema. Konkretno, istraženo je kako se poboljšava performansa igara pomoću optimizacije, kako se pripremaju igre za razvoj za mobilne uređaje i kako se treba pripremati generalno za VR razvoj.

ŠTA VAM JE POTREBNO ZA OVU KNJIGU

Da biste studiozno čitali ovu knjigu i izvršili zadatke unutar nje, potrebni su vam Unity 5 softver (koji možete da preuzmete besplatno na adresi <https://unity3d.com>) i odlučnost da uspete! Koristeći ta dva „alata“, možete da naučite da kreirate odlične igre u Unityju.

ZA KOGA JE OVA KNJIGA

Ako ste Unity programer koji želi da kreira i primeni interesantne igre, koristeći nove funkcije Unityja 5.x, onda je ova knjiga za vas. Pretpostavljamo da imate osnovno znanje C# programiranja.

KONVENCIJE

U ovoj knjizi pronaći ćete više različitih stilova za tekst koje smo upotrebili za različite vrste informacija. Evo nekih primera ovih stilova i objašnjenja njihovog značenja.

Reči koda u tekstu, nazivi tabela baze podataka, nazivi direktorijuma, nazivi fajlova, ekstenzije fajla, nazivi putanja, kratki URL-ovi, korisnički unos i Twitter identifikatori su prikazani na sledeći način: „Ovo bi trebalo da bude prevučeno i otpušteno u direktorijum music“.

Blok koda je postavljen na sledeći način:

```
//Collection of combat sounds
public AudioClip[] CombatSounds;
```

Novi termini i **važne reči** su napisani masnim slovima. Reči koje vidite na ekranu, na primer, u menijima ili okvirima za dijalog, biće prikazane u tekstu na sledeći način: „Iz palete Command izaberite opciju **Change Case Lower**“.



Upozorenja ili važne napomene će biti prikazani u ovakvom okviru.



Saveti i trikovi prikazani su ovako.

POVRATNE INFORMACIJE OD ČITALACA

Povratne informacije od naših čitalaca su uvek dobrodošle. Obavestite nas šta mislite o ovoj knjizi – šta vam se dopalo ili šta vam se možda nije dopalo. Povratne informacije čitalaca su nam važne da bismo kreirali naslove od kojih ćete dobiti maksimum.

Povratne informacije nam šaljite e-mailom na adresu informatori@kombib.rs u naslovu poruke napišite naslov knjige.

KORISNIČKA PODRŠKA

Kao ponosnom vlasniku „Packt“ knjige, imamo mnogo štošta da vam ponudimo da bismo vam pomogli da izvučete maksimum iz svoje narudžbine.

Preuzimanje primera koda

Možete da preuzmete fajlove sa primerima koda za ovu knjigu sa vašeg naloga na adresi <http://www.kombib.rs>.

Možete da preuzmete fajlove sa primerima koda, prateći sledeće korake:

1. Posetite veb stranu knjige: <http://knjige.kombib.rs/naucite-unity-5>
2. Kliknite link i preuzmite kod za knjigu ili u pretraživaču unesite: http://www.kombib.rs/preuzimanje/kod/MasteringUnity5x_Code.rar

Kada su fajlovi preuzeti, ekstrahujte direktorijum, koristeći najnoviju verziju:

- ▣ WinRAR / 7-Zip za Windows
- ▣ Zipeg / iZip / UnRarX za Mac
- ▣ 7-Zip / PeaZip za Linux

PREUZIMANJE SLIKA U BOJI ZA OVU KNJIGU

Takođe smo vam obezbedili PDF fajl sa kolor snimcima ekrana/dijagrama koji su upotrebljeni u ovoj knjizi. Kolor slike će vam pomoći da bolje razumete promene u ispisu. Možete da preuzmete ovaj fajl sa adrese; https://www.packtpub.com/sites/default/files/downloads/MasteringUnity5x_ColorImages.pdf.

ŠTAMPARSKE GREŠKE

Iako smo preduzeli sve mere da bismo obezbedili tačnost sadržaja, ipak su moguće greške. Ako pronađete grešku u nekoj od naših knjiga (u tekstu ili u kodu), bićemo vam zahvalni ako nam to prijavite. Na taj način možete da poštedite druge čitaoce od frustracije, a nama da pomognete da poboljšamo naredne verzije ove knjige. Ako pronađete neku štamparsku grešku, molimo vas da nas obavestite, tako što ćete nam poslati imejl na adresu informatori@kombib.rs

PIRATERIJA

Piraterija autorskog materijala na Internetu je aktuelan problem na svim medijima. Mi u „Packtu“ i @Kompjuter biblioteci@ zaštitu autorskih prava i licenci shvatamo veoma ozbiljno. Ako pronađete ilegalnu kopiju naših knjiga na Internetu, u bilo kojoj formi, molimo vas da nas o tome obavestite i da nam pošaljete adresu lokacije ili naziv web sajta da bismo mogli da podnesemo tužbu.

Molimo vas, kontaktirajte sa nama na adresi copyright@packtpub.com ili informatori@kombib.rs i pošaljite nam link ka sumnjivom materijalu.

Zahvalni smo vam na pomoći u zaštiti naših autora.

PITANJA

Ako imate bilo kakvih problema sa bilo kojim aspektom knjige, možete da kontaktirate sa nama na adresi informatori@kombib.rs - mi ćemo učiniti sve što je u našoj moći da rešimo svaki problem.



1

Priprema i konfigurisanje elemenata postavke

Pred vama je sveobuhvatno i detaljno istraživanje Unityja 5, u kojem se pažljivo ispituje kako se projektuje igra od početka do kraja. U ovoj knjizi ćemo obratiti posebnu pažnju na najbolju praksu, elegantnost dizajna i tehničku izuzetnost. Projekat koji ćemo kreirati će biti FP bioskopska pucačka igra za desktop računare i mobilne uređaje, inspirisana igrom *Typing of the Dead* (https://en.wikipedia.org/wiki/The_Typing_of_the_Dead). Igru ćemo nazvati *Dead Keys* (skraćeno - DK). U njoj igrač se neprestano suočava sa zlim zombijima mesožderima i jedini način da ih bezbedno eliminiše je da izvrši set vežbi kucanja, koristeći fizičku ili virtuelnu tastaturu. Svaki zombi, kada se pojavi, može da napadne igrača i povezan je sa jednom rečju ili frazom nasumično izabranom iz rečnika. Izabrana fraza je jasno predstavljena kao GUI naslov iznad glave zombija. Kao odgovor, igrač mora da, bez greške, ukuca odgovarajuću reč ili frazu bez greške, slovo po slovo, da bi uništio zombija. Ako igrač pogrešno unese reč, na primer - ako pritisne pogrešno slovo ili pogreši u redosledu slova, mora da ponovi sekvencu kucanja od početka.

Ovaj izazov možda zvuči jednostavno za igrača, ali što je reč ili fraza duža, zombi ima više vremena i više mogućnosti da napadne igrača. Igrač ima ograničeno zdravlje i umreće ako zdravlje spadne ispod 0. Prema tome, cilj igrača je da savlada sve zombije i stigne do kraja nivoa.



Dead Keys, igra koju ćemo kreirati

Kreiranje projekta pucanja rečju uključuje mnoge tehničke izazove i 3D i 2D, što podrazumeva široku upotrebu Unityja i njegovog bogatog seta funkcija. Zbog toga, вреди da utrošimo malo vremena u istraživanju onoga što ćemo videti u ovoj knjizi. Reč „*naučite*“ u naslovu knjige „*Naučite Unity 5*“ ukazuje da su veoma važna očekivanja čitalaca o izuzetnosti i složenosti. Ova očekivanja veoma variraju, zato što ljudi imaju različite ideje o tome šta, u stvari, znači sugestija „*naučite*“. Neki misle da je to učenje specifične veštine u kojoj će postati veoma dobri, kao što je učenje programiranja, osvetljenja ili animacije. Naravno, i to su legitimna razumevanja učenja. Međutim, neki drugi korisnici na učenje gledaju kao na celinu - i ova tačka gledišta je legitimna. Ideja autora ove knjige je da se učenje sastoji od sticanja generalnog, sveobuhvatnog znanja više različitih veština i disciplina, ali na specijalan način, u kojem vidimo odnos između njih i gledamo ih kao komplementarne delove koji funkcionišu zajedno da bi proizveli sofisticirane i majstorske rezultate. Ovo je drugo i podjednako legitimno razumevanje termina „*naučiti*“ i ono predstavlja osnovu za ovu knjigu.

U ovoj knjizi se govori o upotrebi Unityja, generalno, kao sveobuhvatne alatke, pregledom mnogih njenih funkcija koje dolaze zajedno, kao jedna jedinica, od editovanja nivoa i programiranja, do osvetljenja, dizajna i animacije. Zbog toga, naš put će nas neminovno dovesti do mnogih područja razvoja – ne samo kodiranja. Prema tome, ako tražite knjigu koja je posvećena samo kodiranju, potražite „*Mastering Unity Scripting*“, čiji je izdavač „*Packt*“. U svakom slučaju, ova knjiga govori o učenju i neće se fokusirati na osnovne koncepte i osnovne operacije. Pretpostavljamo da možete da izgradite osnovne nivoe pomoću editora nivoa i da možete da kreirate osnovne materijale i neke osnovne fajlove skripta pomoću C#-a. Iako će ova knjiga ponegde uključivati neke dodatne, osnovne informacije kao podsetnik i dodaje kontekst, nećemo detaljno opisivati osnovne koncepte, koji su opisani u drugim knjigama („*Packtove*“ knjige koje su za početnike su „*Unity 5.x By Example*“, „*Learning C# by Developing Games with Unity 5.x*“ i „*Unity Animation Essentials*“). Pretpostavljamo da imate osnovno znanje o Unityju i da želite da nadgradite svoje veštine na viši nivo, razvijajući veštinu izgradnje Unity igara.

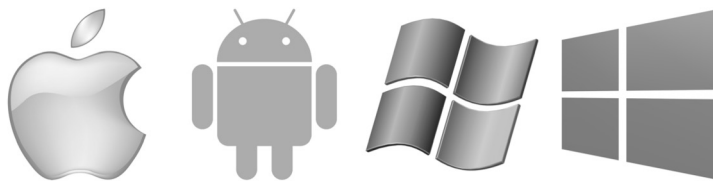
JASAN DIZAJN

Da bismo profesionalno kreirali igru i maksimizirali produktivnost, uvek treba da razvijamo igru iz jasnog dizajna, bez obzira da li je dizajn na papiru ili u digitalnoj formi. Uverite se da je dizajn postavljen i izražen na način koji je razumljiv i drugima, a ne samo vama. Veoma je lako da bilo ko uzbuđeno uskoči u Unity bez plana dizajna, pod pretpostavkom da dobro poznaje svoj um, a zatim da se nađe u situaciji kako besciljno luta od opcije do opcije, bez ikakvog cilja. Bez jasnog plana, projekat će brzo zapasti u kaos. Stoga, prvo kreirajte jasan dokument dizajna igre (**game design document – GDD**) za opštu publiku dizajnera igre koji možda ne poznaju tehnički deo razvoja. U tom dokumentu biće vam jasni neki veoma važni detalji pre nego što upotrebite softver za programiranje, kreirate elemente ili izgradite nivoe. Ovi detalji i opisi su izlistani u sledećim odeljcima, zajedno sa primerima koji se primenjuju na projekat koji gradimo.



GDD je pisani dokument koji kreiraju dizajneri - u njemu je detaljno (pomoću reči, dijagrama i slika) i jasno opisana cela igra. Više informacija o GDD-u možete da pronađete na adresi https://en.wikipedia.org/wiki/Game_design_document.

Ciljne platforme



Ciljne platforme

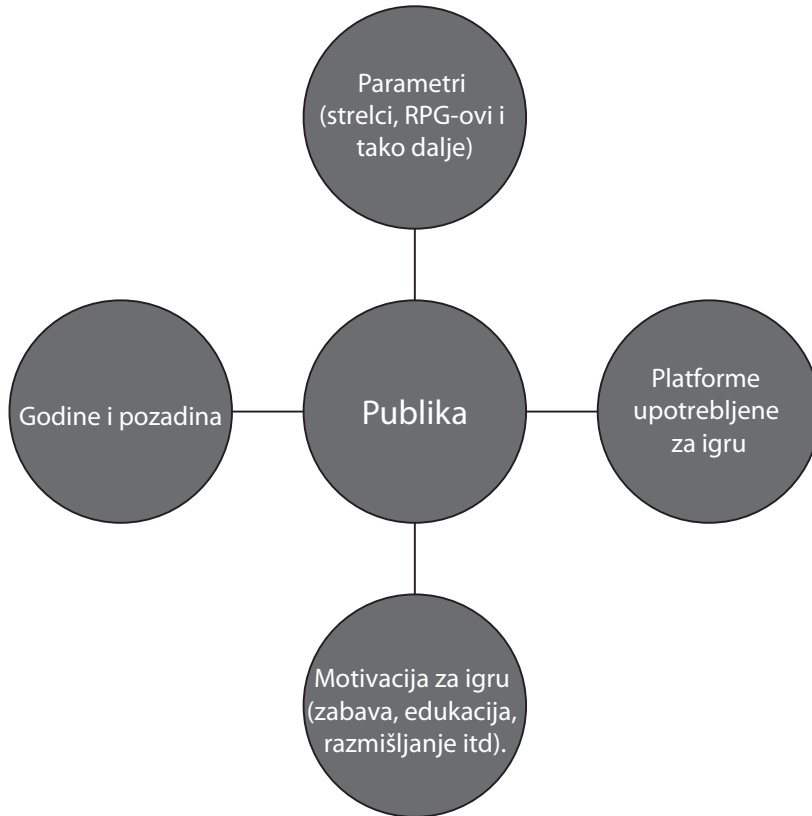
Ciljne platforme specifikuju uređaj ili raspon uređaja na kojima će se igra izvorno pokretati, kao što su Windows, Mac, Android, iOS i tako dalje. Ovo je ceo raspon hardvera na kojima potencijalni igrač može da igra našu igru. Ciljne platforme za DK uključuje Windows, Mac, Android i iOS.

Donošenje odluke o platformi koja će biti podržana je veoma važno logističko i tehničko pitanje. Obično, programer želi da podrži što više platformi, čineći svoju igru dostupnom većem krugu korisnika. Međutim, generalna podrška za svaku platformu je skoro neizvodljiva i zbog toga moraju da se izvrše praktični izbori. Svaka podržana platforma uključuje određeno vreme, napor i novac programera, čak i ako Unity olakšava višeplatformsku podršku izvršavanjem lakših zadataka. Kreiranje igre za više platformi normalno znači kreiranje mrežica, tekstura i audio fajlova različitih veličina i nivoa detalja, kao i prilagođavanje korisničkih interfejsa za različite rasporede ekrana i razmere, a takođe treba da obratimo pažnju i na hardverske specifičnosti svake platforme.

Podrška za platformu utiče i na osnovnu mehaniku igre; na primer, igre za ekrane osetljive na dodir se ponašaju drugačije od igara zasnovanih na tastaturi, a kontrole pokreta se razlikuju od kontrole zasnovanih na mišu. Prema tome, platforma uvek sputava i ograničava mogućnosti na ono što može da se uradi, i to ne samo tehnički, već i za sadržaj. App Store smernice za podnošenje sadržaja postavljaju stroge zahteve za dozvoljeni sadržaj, jezik i reprezentacije igara i omogućavaju kupovinu unutar aplikacije i pristup eksternim sadržajima koje je kreirao korisnik.

Zaključak je da bi ciljne platforme trebalo, uglavnom, da budu izabrane unapred. Ta odluka će u velikoj meri uticati na osnovnu mehaniku igre i način na koji je dizajn implementiran. Ponekad, odluka za odlaganje podrške za određenu platformu može (i trebalo bi) da bude doneta zbog određenih tehničkih i ekonomskih razloga. Međutim, kada je doneta takva odluka, imajte na umu da ona može u velikoj meri da produži vreme razvoja u daljem ciklusu, jer će možda biti potrebni podešavanja i ponovno programiranje za pravilnu podršku nijansi platforme.

Publika za koju se nešto projektuje



Odlučivanje o publici za koju se nešto projektuje

Publika za koju se nešto projektuje je kao profil ličnosti. Upotrebom nekih stereotipa specifikuje se ko bi trebalo da igra našu igru: povremeni igrači, igrači akcionih igara ili „zagriženi“ igrači, deca ili odrasli, oni koji govore engleski ili koji ne govore engleski ili neko drugi. Ovo je važno, posebno za utvrđivanje pogodnosti sadržaja, karaktera i težine igranja. Pogodnost nije samo skrivanje golotinje, nasilja i psovki od mlađih igrača, već je reč o povezivanju publike sa relevantnim sadržajem: sa temama, pričama i idejama koje su im interesantne i koje će ih ohrabriti da nastave da igraju igru. Slično tome, težina igre ne govori o tome da se igra pojednostavi za mlađe igrače, već je reč o balansiranju nagrada, kazni i tajminga za odgovor na očekivanja publike, bez obzira na njihove godine.

Kao platformu za koju se kreira igra, treba da imate na umu i publiku kada projektujete igru. To je posebno važno da biste ostali fokusirani kada uključujete nove ideje u igru. Smišljanje novih ideja je odlično, ali da li će one funkcionisati za publiku? Ako publika za koju se nešto projektuje nije dovoljno fokusirana, mogu se javiti neki od sledećih problema:

- Igra će biti konceptualno neuredna (zbrkane, nepovezane ideje).
- Biće vam teško da odgovorite na koji je način igra zanimljiva.
- Neprestano ćete izvršavati velike i važne promene u dizajnu u toku razvoja.

Zbog ovih i drugih razloga, suzite publiku za koju se nešto projektuje što je moguće preciznije i što je moguće ranije.

Za igru *Dead Keys* publika za koju se nešto projektuje će biti korisnici stariji od 15 godina i obožavaoci igre *Shoot, Em Up*, koji, takođe, uživaju u provokativnim igrama koje odstupaju od mejnstrima. Ostala publika može da uključuje povremene igrače koji uživaju u igrama reči vremenski ograničenim.

Žanr

Žanr je, pre svega, sadržaj igre. Da li je RPG, FPS, avanturistička ili nekog drugog tipa? Žanr može da bude još uže definisan, kao što je fantasy MMORPG i cyberpunk, competitive, deathmatch i first-person-shooter. Ponekad ćete želeti da on bude veoma specifičan, a ponekad ne, u zavisnosti od vašeg cilja. Kada gradite igru, budite specifični kada utvrđujete žanr što je moguće tačnije. U ovom slučaju ideja je da obavimo *dobar posao*, koristeći isprobanu i testiranu formulu. Nasuprot tome, izbegavajte previše uske definicije kada tražite inovacije i pomerate granice. Slobodno kombinujte postojeće žanrove na nove načine ili, ako stvarno želite izazov, osmislite potpuno novi žanr.

Inovacija može da bude zanimljiva, ali i veoma rizična. Lako ćete pomisliti da je vaša najnovija ideja pametna i ubedljiva, ali uvek se konsultujte sa drugim ljudima da biste videli njihove reakcije i naučili da prihvatite konstruktivnu kritiku već u ranoj fazi razvoja. Zatražite od drugih korisnika da isprobaju igru koju ste kreirali ili da isprobaju prototip igre na osnovu dizajna. Međutim, nemojte se mnogo oslanjati na dizajn zasnovan na dokumentima kada procenjujete zabavu i mogućnost igranja, jer se iskustvo igranja potpuno razlikuje od čitanja i misli koje čitanje generiše.

Za igru *Dead Keys* žanr će biti cinematic first-person zombie-typer. U ovoj igri žanr koristi postojeću i dobro poznatu tradiciju first-person shooter, ali (radi inovacije) zamenjuje definisanje elementa pucanja kucanjem.

Režim igre

Termin „*režim igre*“ može da ima više značenja, ali u ovom slučaju ćemo se fokusirati na razliku između režima igre jednog igrača i više igrača. Dead Keys će biti igra za jednog igrača, ali u dizajnu igre ne postoji ništa što ukazuje da je to igra samo za jednog igrača. Ona može da se prilagodi i za lokalni co-op multiplejer i za internet multiplejer (pomoću mrežnih funkcija Unityja). Više informacija o Unity mreži možete da pronađete na adresi <https://docs.unity3d.com/Manual/UNet.html>.

Važno je da za tehničke probleme nađete odgovarajuća rešenja veoma rano u razvoju, pošto će to imati veliki uticaj na način na koji će igra biti konstruisana i na funkcije koje će ona podržavati.

Cilj u igri

Svaka igra (osim eksperimentalne i iskustvene) zahteva cilj za igrača; nešto što igrač treba da uradi, ne samo unutar specifičnih nivoaa, već u celoj igri uopšte. Ovaj cilj je važan ne samo za igrača (da bi igra bila zanimljiva), već i za programera da proceni koliko izazova, raznolikosti i interesovanja može da se doda u kombinaciju. Pre nego što započnemo kreiranje igre, treba da imamo na umu jasno definisan cilj igre.

Izazovi su uvedeni primarno kao prepreke za cilj, a bonusi, u stvari, olakšavaju postizanje cilja. Za igru Dead Keys primarni cilj je da „preživimo“ i stignemo do kraja nivoaa. Zombiji su pretnja tom cilju, zato što napadaju i uništavaju igrača, a bonusi postoje da bi igra bila interesantnija.



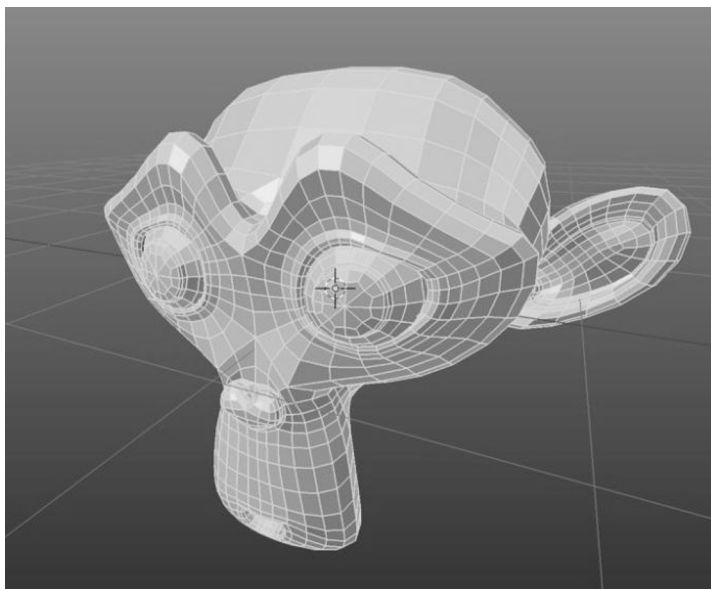
Preporučujem da upotrebite alatke za upravljanje projektom i za međusobnu saradnju tima za kreiranje grafikona i dokumenata i da vremenski pratite zadatke unutar projekta. To možete da uradite i besplatno; neke od online alatki koji će vam pomoći su Trello (<https://trello.com>), Bitrix24 (<https://www.bitrix24.com>), Basecamp (<https://basecamp.com>), Freedcamp (<https://freedcamp.com>), Unfuddle TEN (<https://unfuddle.com>), Bitbucket (<https://bitbucket.org>), Microsoft Visual Studio Team Services (<https://www.visualstudio.com/en-us/products/visual-studio-team-services-vs.aspx>) i Concord Contract Management (<http://www.concordnow.com>).

PRIPREMA ELEMENATA

Kada imamo jasnu odluku o početnom konceptu i projektu, spremni smo za prototip, odnosno za izgradnju Unity projekta koji prikazuje osnovnu mehaniku i pravila igre u akciji kao primer koji može da se igra. Nakon toga, obično ćemo izvršiti neka manja podešavanja u dizajnu i ponoviti kreiranje prototipa, dok ne dođemo do artefakta koji želimo da pratimo. Tada umetnički tim mora da proizvede elemente (mrežice i teksture) na osnovu koncepta, dizajna igre i fotografskih referenci. Kada proizvodimo mrežice i teksture za Unity, treba da pratimo neke važne smernice da bismo postigli optimalnu grafičku performansu u igri. To su smernice o strukturiranju i izgradnji elemenata na pametan način da bi lako mogli da se eksportuju iz njihovog originalnog softvera, a zatim da budu lako importovani dok se izvršavaju najbolje što mogu pri pokretanju. Hajde da pogledamo neke od ovih smernica za mreže i teksture.

Mrežice – koristite samo dobru topologiju

Dobra topologija mrežice se sastoji od mnogouglova koji imaju tri ili četiri stranice u modelu (ne više). Osim toga, Edge Loops bi trebalo da teče na uređeni, regularan način, duž kontura modela, definišući njegov oblik i formu.

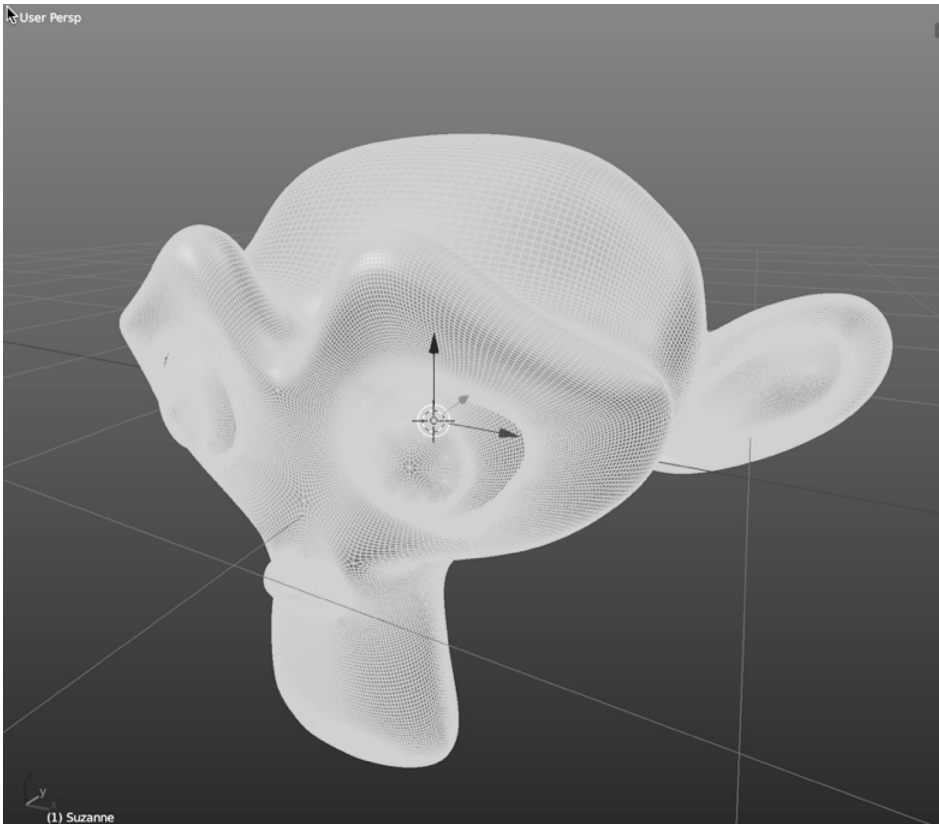


Jasna topologija

Pri importovanju Unity automatski konvertuje sve **NGons** (mnogouglove sa više od četiri stranice), ako ih mrežica ima. Međutim, bolje je da izgradite mrežice bez tih mnogouglova, umesto da se oslanjate na automatizovane metode Unityja. Zahvaljujući tome, ne samo da ćete steći dobre navike u fazi modelovanja, već ćete izbeći automatsku i nepredvidivu topologiju mrežice, što utiče na način kako je ona osenčena i animirana.

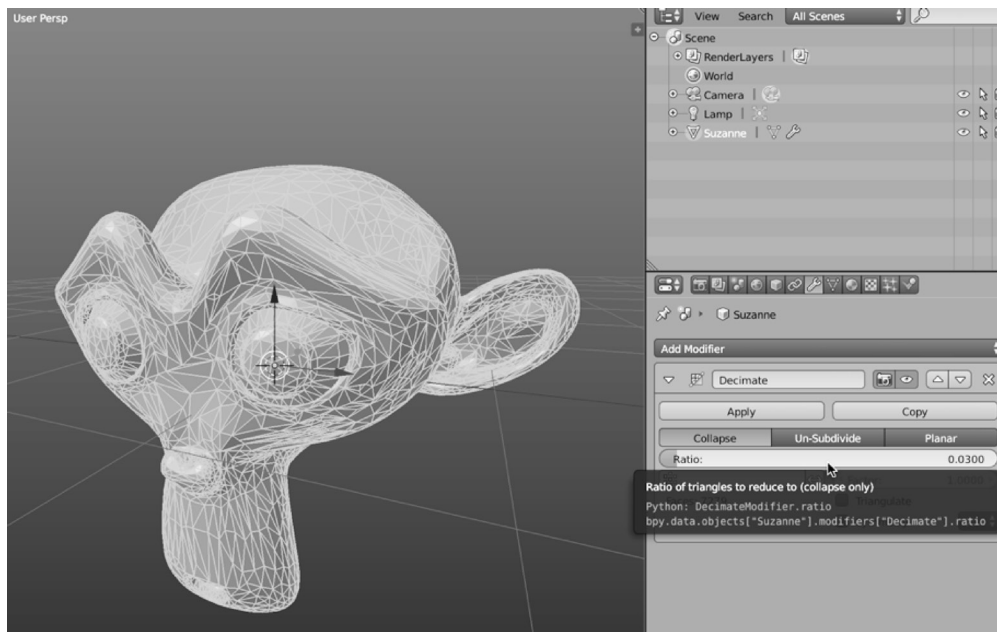
Mrežice – minimizirajte broj poligona

Svaki mnogougao u mrežici zahteva onoliku performansu renderovanja koliko je vremena potrebno GPU-u da obradi i renderuje svaki mnogougao. Prema tome, razumno je minimizirati broj mnogougao u mrežici, čak i ako je moderan grafički hardver vešt u korišćenju više mnogougao. Najbolje je da smanjimo broj mnogougao kada god je to moguće i u meri koja neće umanjiti našu umetničku viziju i stil.



Mrežice sa puno mnogougao (pokušajte da smanjite broj mnogouglova gde god je to moguće)

Dostupne su mnoge tehnike za smanjivanje broja mnogougona. Većina 3D aplikacija (kao što su 3ds Max, Maya i Blender) pruža automatizovane alate za desetakovanje mnogougona u mrežici, dok se zadržavaju osnovni oblik i okvirna linija. Međutim, ovi metodi često naprave kaos u topologiji, ostavljajući isturene površine i ivice u svim pravcima. To može, čak, da bude korisno za smanjivanje mnogougona u **statičnim mrežicama** (mrežice koje se nikad ne animiraju), kao što su statue, kuće ili stolice, ali je loše za animirane mrežice u kojima je topologija posebno važna.



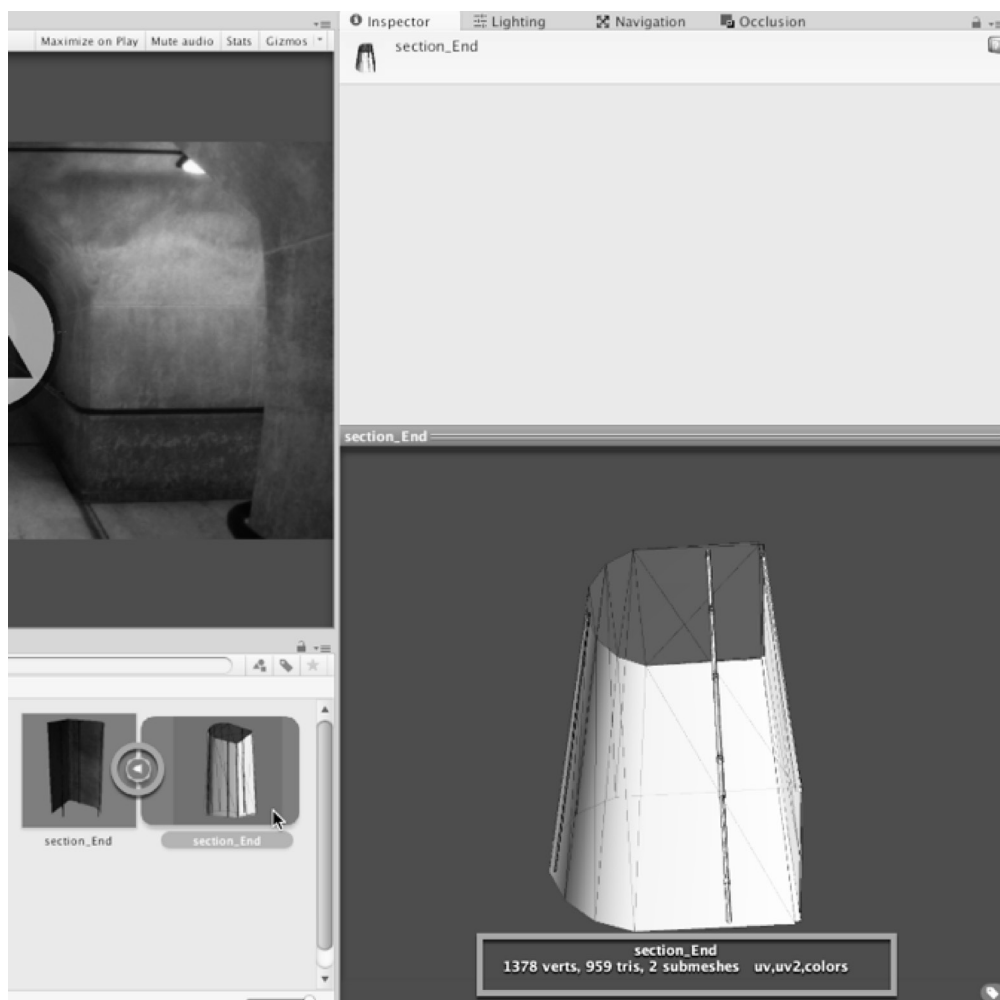
Smanjivanje mnogougona mrežica pomoću automatizovanih metoda može da izazove neurednu topologiju!



Ako želite da znate ukupan broj temena i površina u mrežici, možete da upotrebite statistiku 3D softvera. Blender, Maya, 3ds Max i većina 3D softvera omogućavaju da vidite broj temena i površina selektovanih mrežica direktno iz prozora za prikaz. Međutim, ovu informaciju treba razmatrati samo kao grubu smernicu, jer, kada importujemo mrežicu u Unity, često je broj temena veći od očekivanog. Postoji mnogo razloga za to, što je i detaljno opisano na adresi <http://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html>.



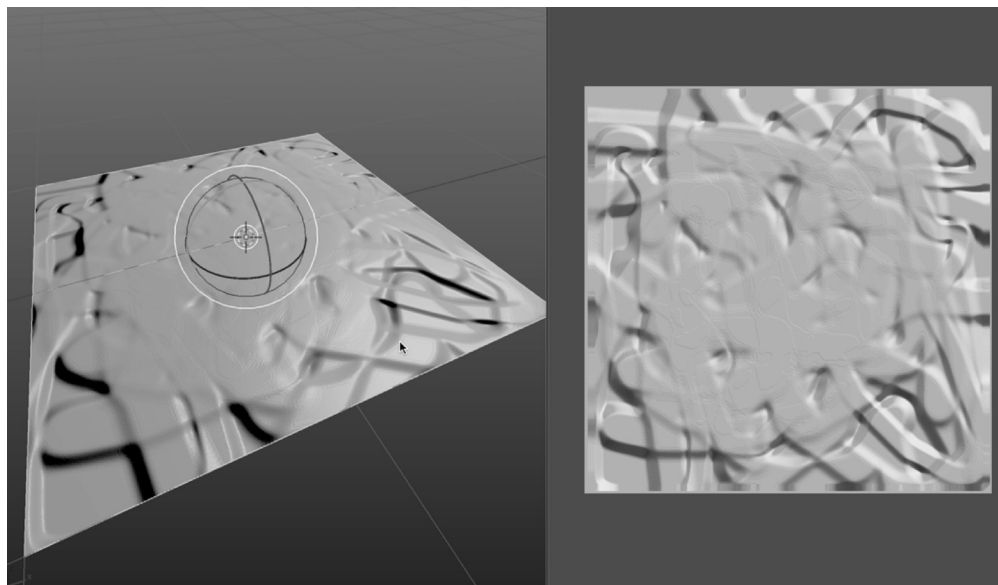
Ukratko, upotrebite broj temena u Unityju kao stvarni broj temena u mrežici. Da biste pregledali broj temena za importovanu mrežicu u Unityju, kliknite na strelicu udesno na umanjenom prikazu mrežice na panelu **Project** da bi bili prikazani interni elementi mrežice. Selektujte ovaj element, pa prikažite broj temena u panelu prikaza u objektu **Inspector**.



Prikaz broja temena i površina za mrežice u Unityju

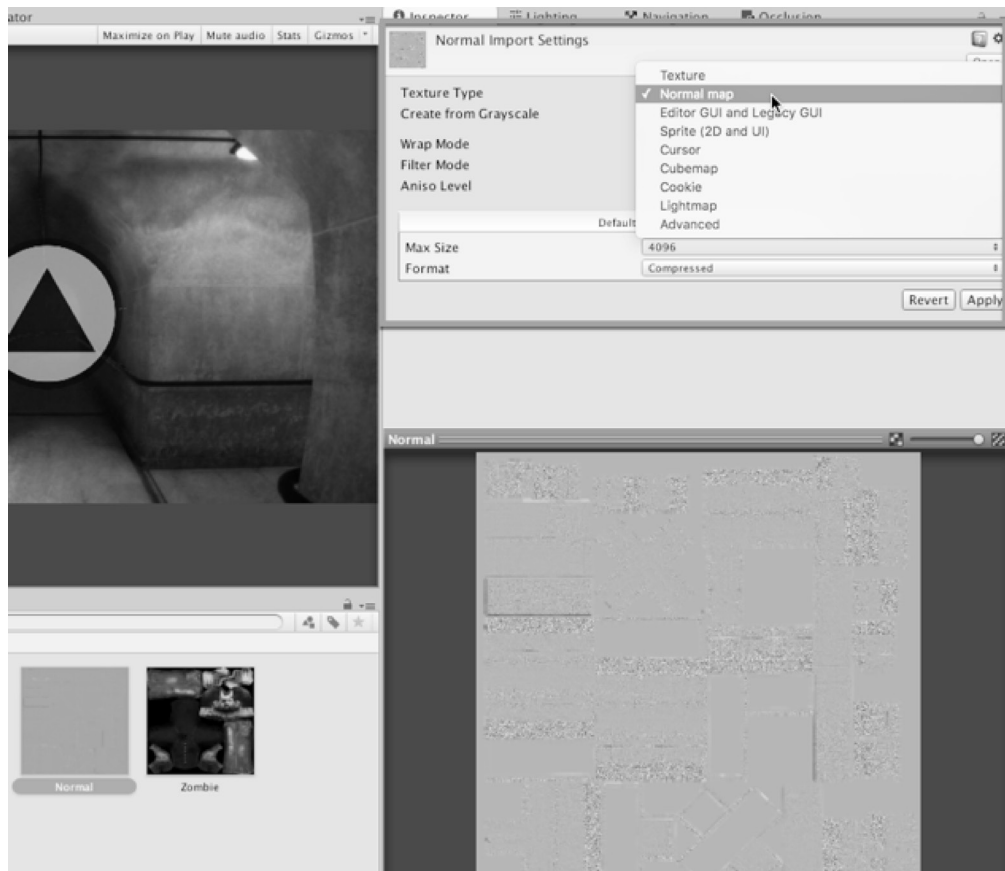
Mrežice – simuliranje detalja reljefa bez geometrije

Kao što smo pomenuli, pokušajte da zadržite što manji broj mnogougona u mrežici. Međutim, mrežice sa malo mnogougona su nižeg kvaliteta od mrežica visoke rezolucije. Sadrže manje mnogougona i, stoga, imaju manje detalja. Ipak, to ne bi trebalo da predstavlja problem. Postoje tehnike za simuliranje detalja u mrežicama sa manje mnogougona, koje čine da se ove mrežice prikažu u višoj rezoluciji nego što je realno. Normal Mapping je jedan primer. **Normal Maps** su specijalne teksture koje definišu orijentaciju i grubost površine mrežice za njene mnogougone i način na koji ti mnogougoni vrše interakciju sa osvetljenjem. Ukratko, Normal Map specifikuje kako će osvetljenje biti na mrežici i utiče na to kako je mrežica osenčena. To utiče na način na koji mi opažamo detalje. Možemo da proizvedemo Normal Maps na mnogo načina, koristeći, obično, softver za 3D modelovanje. Kreiranjem dve verzije mrežice (konkretno, verzije sa više mnogougona koja sadrži sve potrebne detalje i verzije sa manje mnogougona za primanje detalja) možemo da primenimo normalnu informaciju iz mrežice sa više mnogougona u mrežici sa manje mnogougona pomoću fajla teksture. Ovaj pristup (poznat kao **Normal Map Baking**) može da dovede do neverovatno preciznih i uverljivih rezultata, kao što su sledeći:



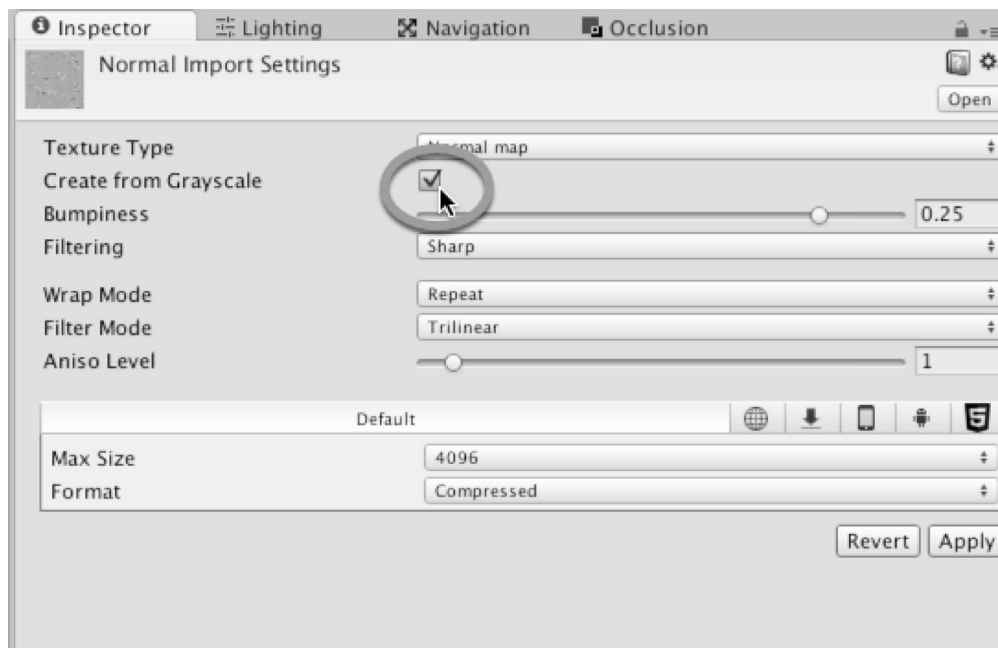
Simuliranje detalja više mnogougona pomoću Normal Mapsa

Međutim, ako nemate Normal Maps za importovane mrežice, Unity može da ih generiše iz standardne, difuzne teksture, koristeći podešavanje **Import Settings**. Korišćenje ovog načina možda neće dovesti do najuverljivijih i fizički tačnih rezultata kao Normal Map Baking, ali je korisno ako želimo da brzo i lako generišemo detalje pokreta, poboljšavajući atmosferu i realizam scene. Da bismo kreirali Normal Map iz difuzne teksture, prvo ćemo selektovati importovanu teksturu na panelu **Project** i dupliraćemo je – uverićemo se da originalna verzija nije uništena ili promenjena. Zatim, iz objekta **Inspector** promenićemo opciju **Texture Type** (za dupliranu teksturu) sa **Texture** na **Normal map**. Ova opcija će promeniti način na koji Unity „razume“ i koristi teksturu:



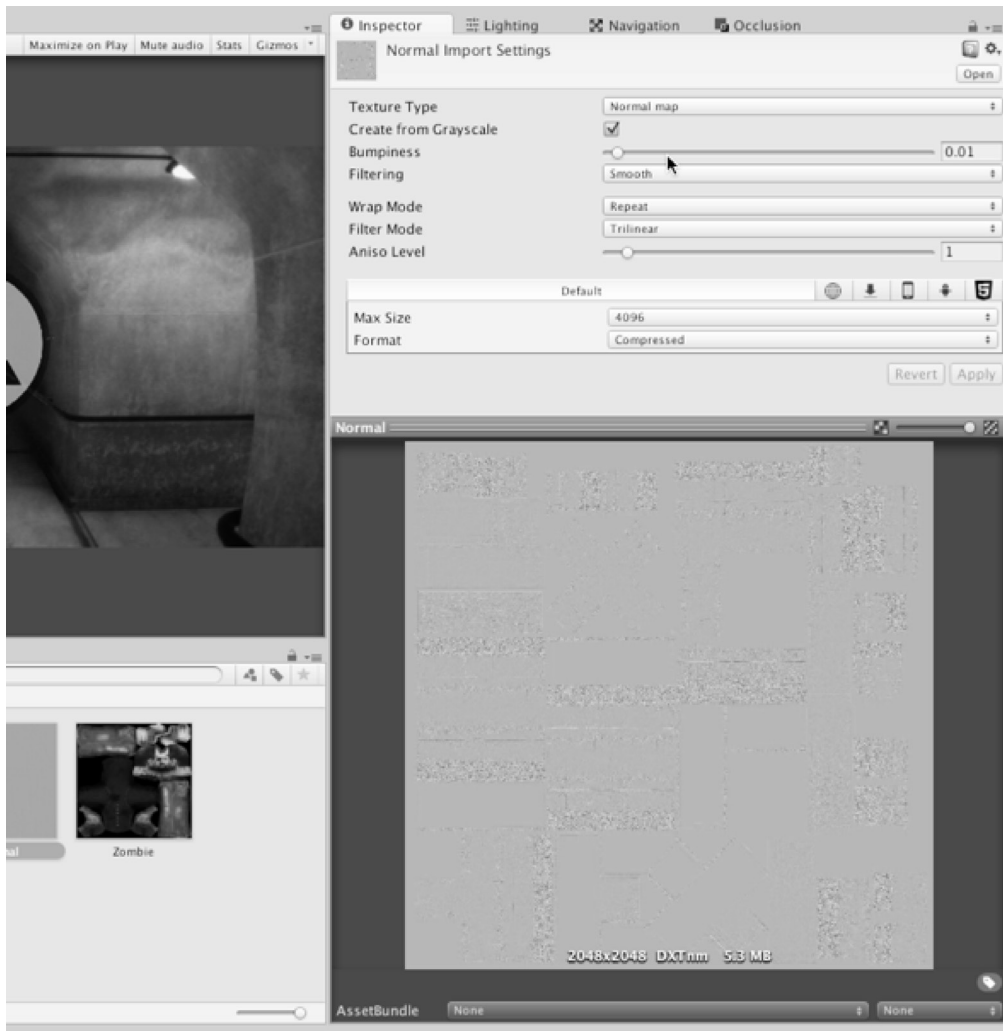
Konfigurisanje teksture kao Normal Map

Specifikovanje Normal Mapa za teksturu konfiguriše Unity za upotrebu teksture na specijalizovan, optimizovan način za generisanje detalja reljefa na modelu. Međutim, kada kreiramo Normal Map iz difuzne teksture, takođe treba da uključimo polje za potvrđivanje **Create from Grayscale**. Kada je ono uključeno, Unity generiše Normal Map iz crno-bele verzije difuzne teksture, koristeći podešavanja **Bumpiness** i **Filtering** na sledeći način:



Uključite Create from Grayscale opciju za Normal Maps

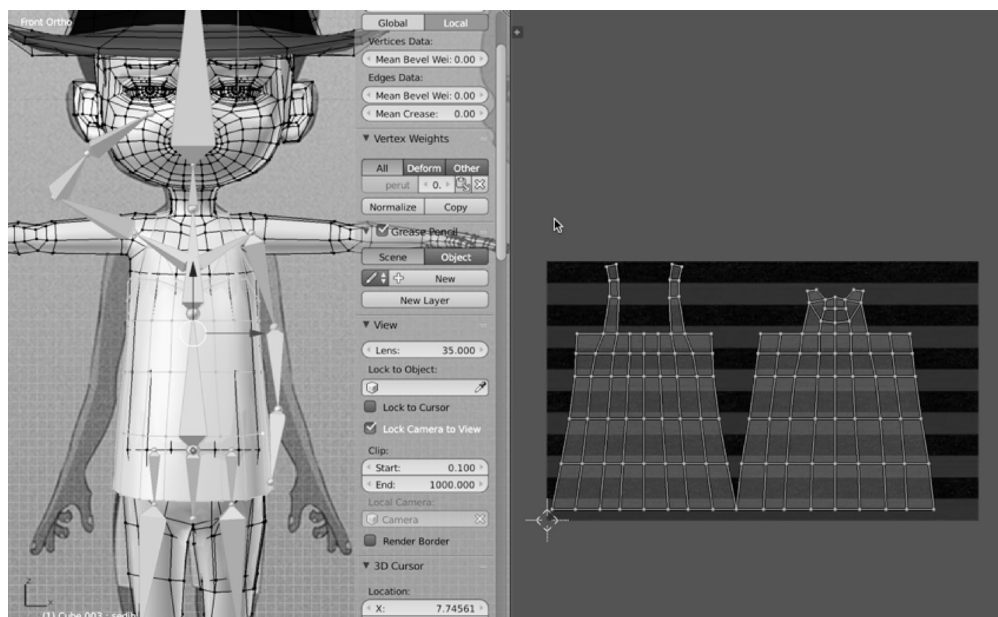
Kada je uključena opcija **Create from Grayscale**, možemo da upotrebimo klizač Bumpiness da bismo pojačali ili smanjili efekat ispupčenja i podešavanje **Filtering** za kontrolu gruboće ili mekoće ispupčenja. Kada podesimo potrebna podešavanja, potvrdićemo promene i prikazati rezultat, tako što ćemo pritisnuti dugme **Apply** u objektu Inspector:



Podešavanje importovanog Normal Mapa

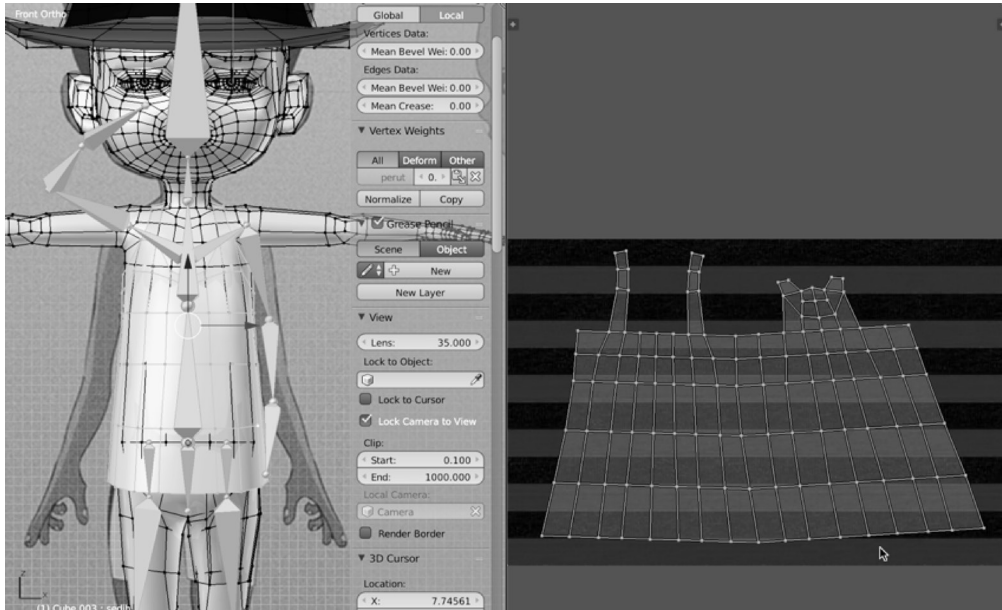
Mrežice – minimiziranje UV šavova

Šavovi su isecci ivica ubačeni u mrežicu u toku UV mapiranja, koji pomažu odvijanje ili ispravljanje mrežice u 2D površinu za dodelu teksture. Ovaj proces se izvršava u softveru za 3D modelovanje, ali isecci koje on kreira su veoma važni za pravilno odvijanje modela da bi model izgledao kao što i treba da izgleda unutar Unityja. Ivica je klasifikovana kao šav u UV prostoru kada ima samo jednu susednu površinu, umesto dve. U suštini, šavovi određuju kako će UV-ovi mrežice biti rasečeni u posebne UV školjke ili UV ostrva, koji su uređeni u finalni UV raspored. Ovaj raspored mapira teksturu na površinu mrežice na sledeći način:



Kreiranje UV rasporeda

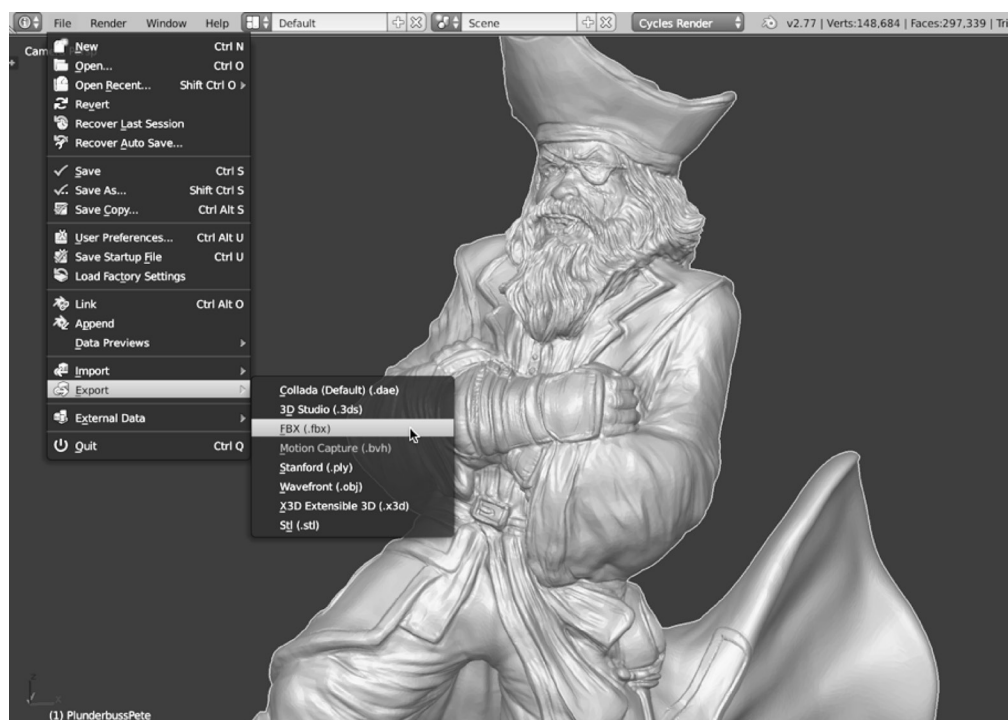
Uvek minimizirajte UV šavove, gde je to izvodljivo, spajanjem neujednačenih ivica, školjki ili ostrva, formirajući veće jedinice. To nije zadatak koji ćemo raditi u Unityju, već u softveru za 3D modelovanje. Na ovaj način ćemo smanjiti i broj temena i složenost mrežice, što će dovesti do poboljšane performanse u Unityju, jer Unity mora da duplira sva temena duž šavova da bi se prilagodio standardima renderovanja za većinu grafičkih hardvera. Prema tome, gde god postoje šavovi, postojaće i dvostruka temena, kao što je prikazano na sledećoj slici:



Spajanje ivica i ostrva radi smanjivanja UV šavova

Mrežice – eksportovanje kao FBX

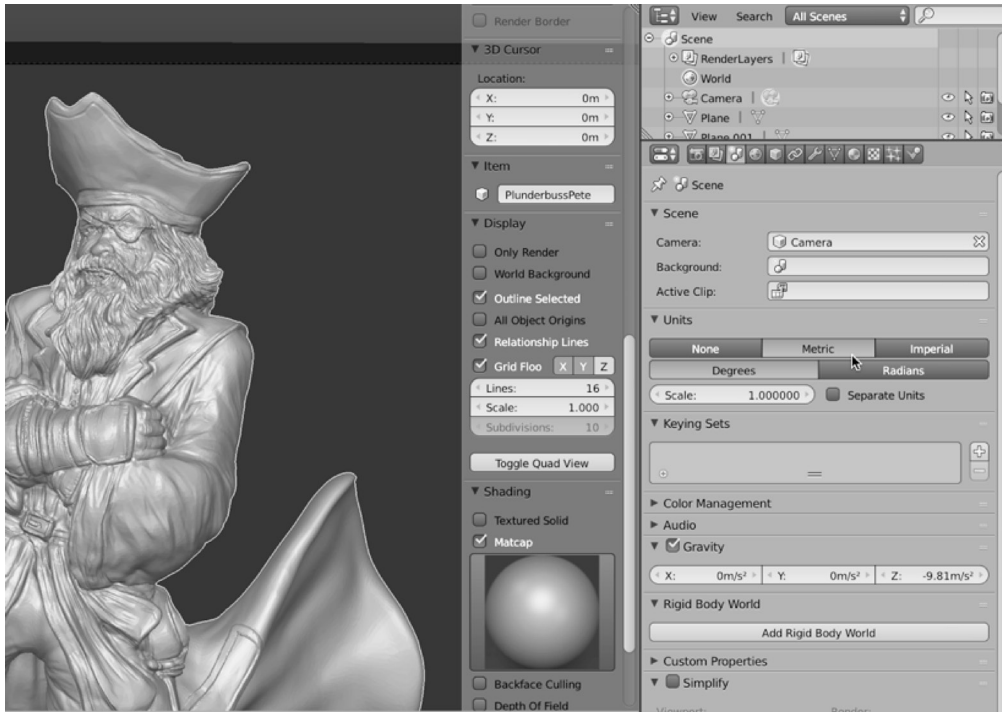
Unity zvanično podržava više formata importovanja mrežica, uključujući .ma, .mb, .max, .blend i druge. Detalje i poređenja ovih formata možete da pronađete na adresi <http://docs.unity3d.com/Manual/3D-formats.html>. Unity deli formate mrežica u dve glavne grupe: na **eksportovane** i na **vlasničke**. Eksportovani formati uključuju .fbx i .dae. To su mrežice koje su ručno eksportovane iz softvera za 3D modelovanje u nezavisni format razmene podataka koji je prepoznat. Funkcije ovog formata su ograničene, ali su široko podržane. Nasuprot tome, vlasnički formati su specifični za aplikacije -podržavaju širi raspon funkcija, ali po cenu kompatibilnosti. Ukratko, skoro uvek treba da koristite eksportovani FBX format fajla. On je najviše podržan, upotrebljen i testiran format unutar Unity zajednice, a podržava importovane mrežice svih tipova, i statične i animirane. Osim toga, ovaj format daje najbolje rezultate. Ako izaberete vlasnički format, često će biti potrebno da importujete dodatne 3D objekte koje nikada nećete upotrebiti u igri, a Unity projekat je automatski „vezan“ za sam 3D softver. Odnosno, biće vam potrebna potpuna licencirana kopija 3D softvera za svaki računar na kojem želite da otvorite Unity projekat. To je neugodno.



Eksportovanje mrežica u FBX fajl funkcioniše najbolje u Unityju.

Mrežice – upotreba metarske (metričke) skale

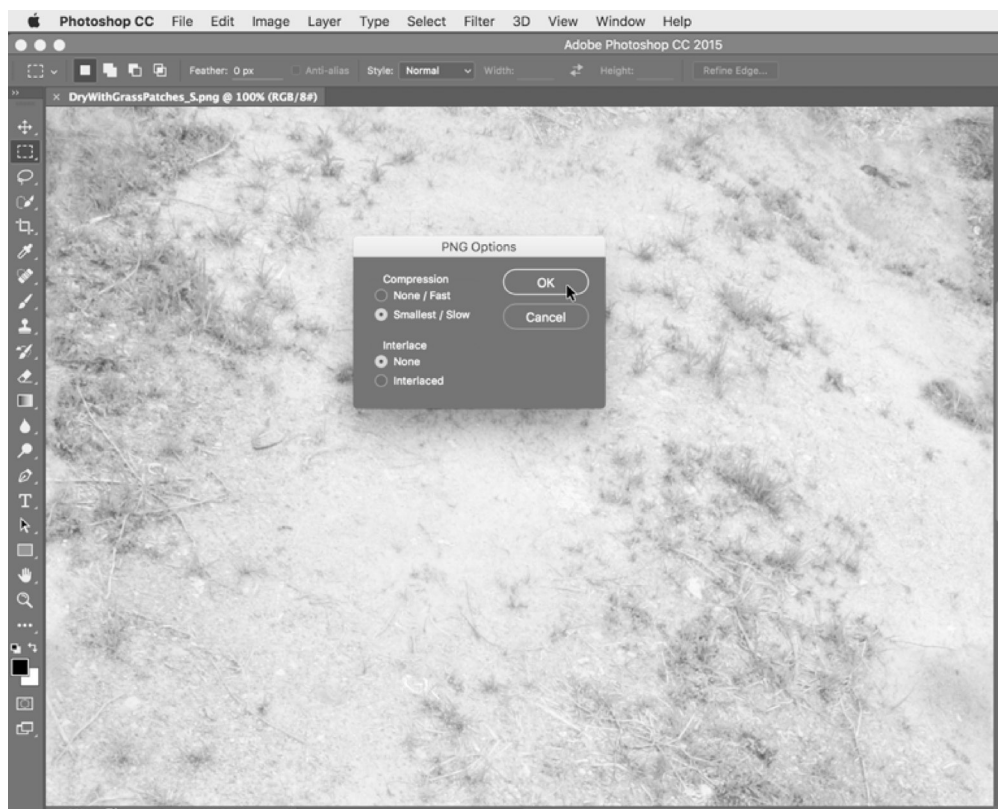
Unity meri 3D prostor pomoću metričkog sistema - sistem fizike jednu svetsku jedinicu „razume“ i „zna“ da to znači jedan metar. Unity je konfigurisan tako da koristi modele iz većine 3D aplikacija, koristeći njihova standardna podešavanja. Međutim, ponekad će modeli biti prikazani previše veliki ili previše mali kada su importovani. To se obično dešava kada svetska jedinica nije konfigurisana na metrički sistem u softveru za 3D modelovanje. Detalji o promeni jedinica variraju za svaki softver, kao što su Blender, Maya ili 3ds Max. Svaki program omogućava podešavanje jedinica iz menija **Preferences**.



Konfigurisanje 3D softvera na Metric jedinice

Teksture – nikada ne koristite kompresije bez gubitka

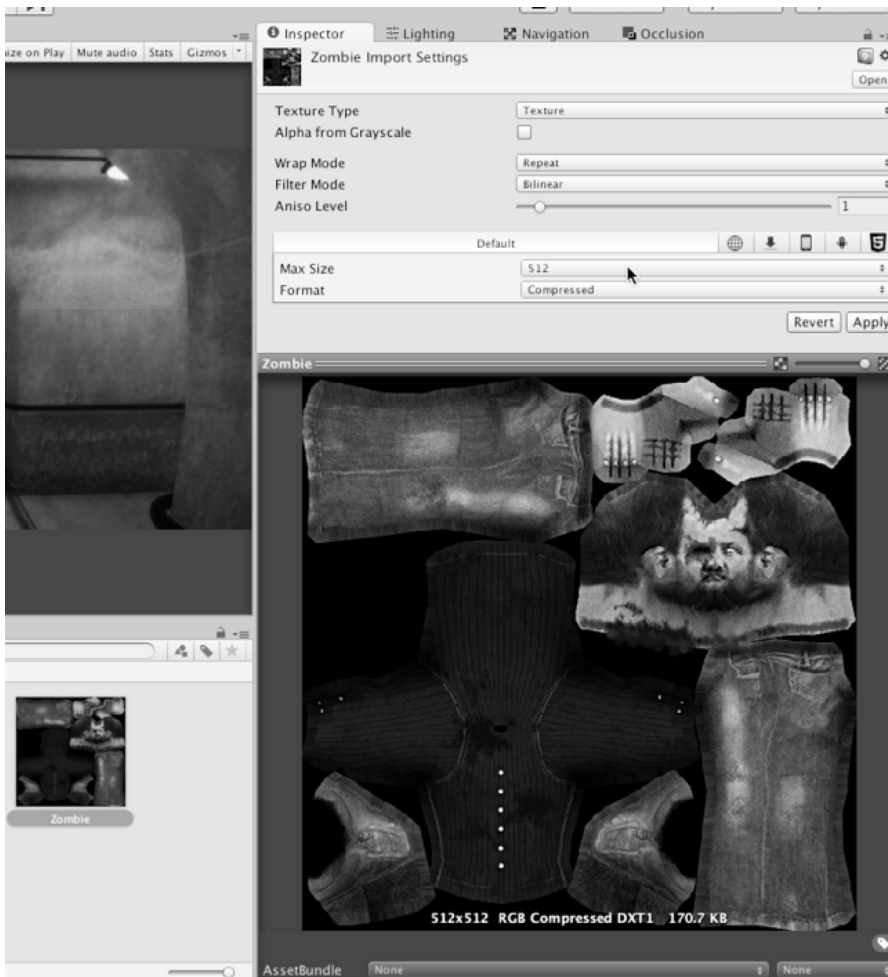
Uvek sačuvajte teksture u formatima bez gubitka, kao što su PNG, TGA ili PSD. Izbegavajte formate sa gubitkom, kao što je JPG, čak i ako oni obično imaju manju veličinu fajla. JPG može da bude idealan za slike za veb sajtove ili za slanje fotografija sa letovanja prijateljima i porodici, ali za kreiranje teksture za video igru ovaj format je problematičan, jer eksponencijalno gubi kvalitet pri svakoj narednoj operaciji snimanja. Kada se upotrebi format bez gubitka i ukloni JPG format iz svakog koraka toka rada (uključujući i posredničke korake), teksture će ostati jasne i oštre:



Snimanje tekstura u PNG fajlove

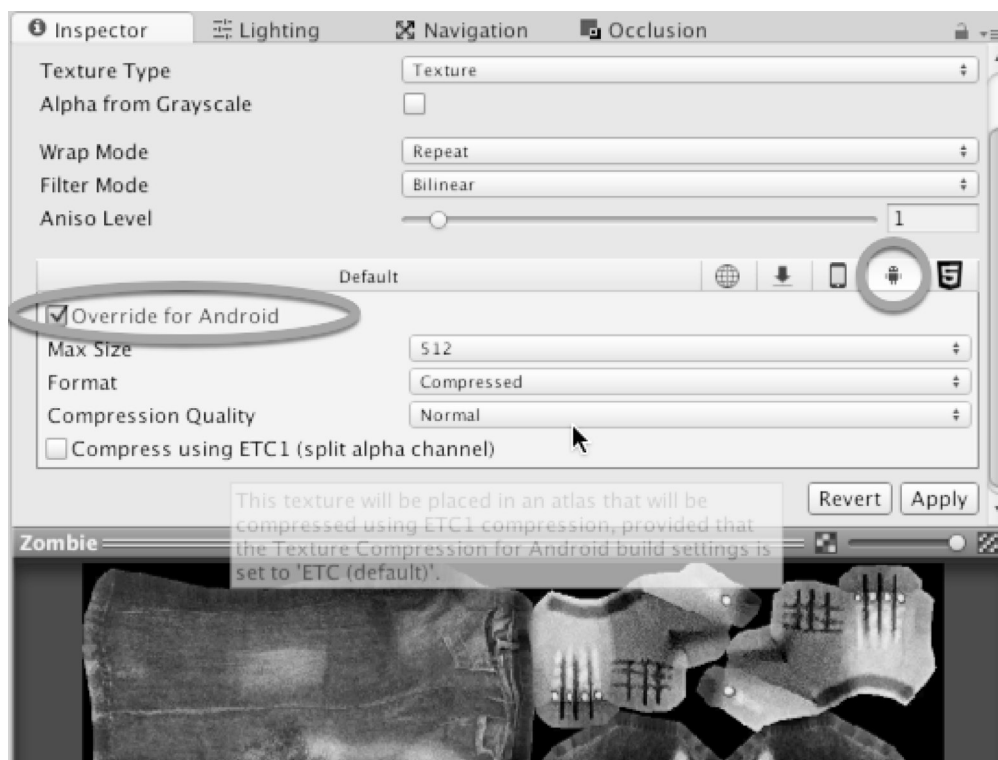
Teksture – power-2 veličine

Ako su teksture za 3D modele i mrežice (ne za sprajtove i GUI elemente), kreirajte njihove dimenzije power-2 veličine za najbolje rezultate. Teksture ne treba da budu kvadratne (jednake širine i visine), ali svaka dimenzija bi trebalo da bude formirana u rasponu power-2 veličine. Validne veličine uključuju 32, 64, 128, 256, 512, 1024, 2048, 4096 i 8192. Dimenzionisanje tekstura na power-2 dimenziju pomaže Unityju da povećava i smanjuje teksturu i da kopira piksele između tekstura po potrebi pomoću najšireg raspona grafičkog hardvera.



Kreiranje tekstura power-2 veličine

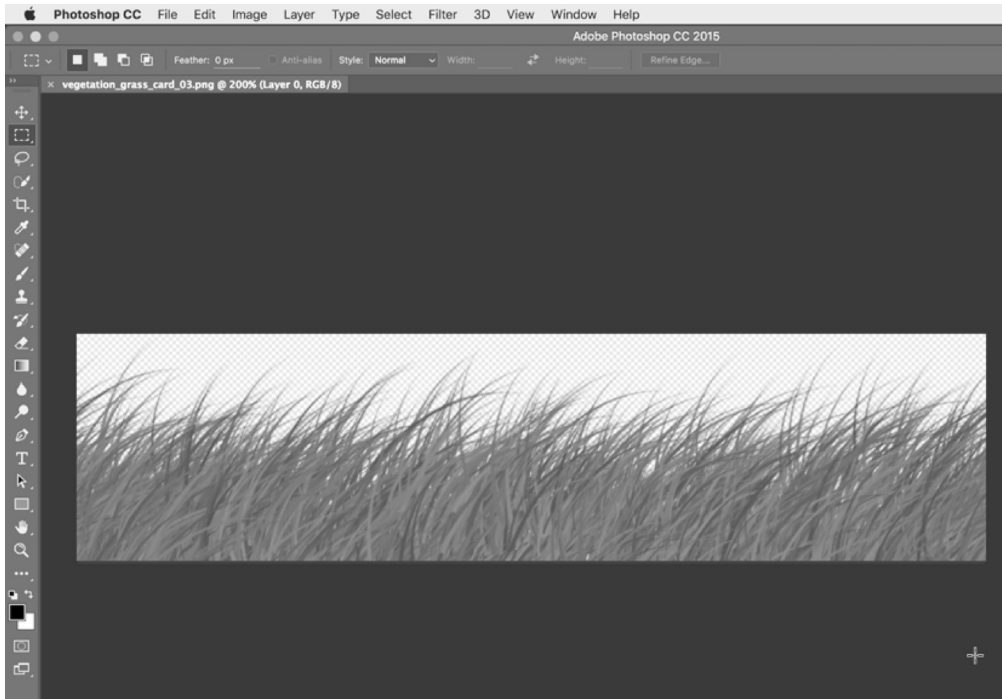
Kada kreiramo teksture, uvek je najbolje da dizajniramo za najveću power-2 veličinu koja će nam biti potrebna (umesto za najveću moguću), a zatim da je smanjimo gde god odgovaraju manje power-2 veličine za stariji hardver i slabije sisteme, kao što su mobilni uređaji. Za svaku importovanu teksturu možemo da upotrebimo Unity platform kartice iz objekta **Inspector** da bismo specifikovali odgovarajuću maksimalnu veličinu za svaku teksturu na specifičnoj platformi: jednu za desktop sisteme, jednu za Android, jednu za iOS i tako dalje. To ograničava maksimalno dozvoljenu veličinu za selektovani cilj na osnovu platforme. Ova vrednost treba da bude najmanja veličina koja je kompatibilna sa umetničkim namerama i namenjenim kvalitetom.



Zamena vrednosti veličina tekture za druge platforme

Teksture – alfa teksture

Alfa teksture su teksture sa providnošću. Kada su primenjene na 3D modele, one čine područja modela providnim, omogućavajući da se prikažu objekti iza modela. Alfa teksture mogu da budu TGA fajlovi sa namenskim alfa kanalima ili PNG fajlovi sa providnim pikselima. U svakom slučaju, one mogu da se renderuju sa artefaktima u Unityju ako nisu pravilno kreirani i importovani.



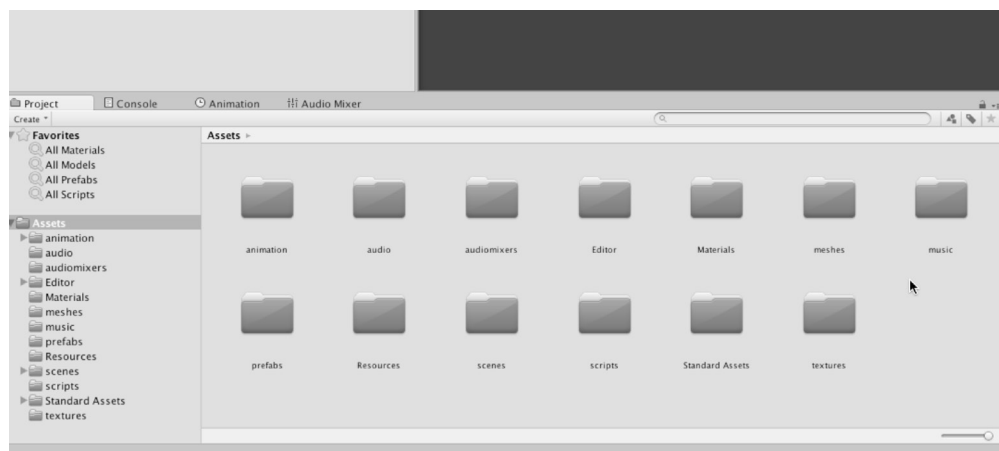
Kriiranje alfa tekstura

Ako treba da upotrebite alfa teksture, uverite se da ste prvo proverili zvaničnu Unity dokumentaciju o načinu eksportovanja ovih tekstura za optimalne rezultate, koju ćete pronaći na adresi <http://docs.unity3d.com/Manual/HOWTO-alphamaps.html>.

IMPORTOVANJE ELEMENATA ZA IGRU DEAD KEYS

U prethodnom odeljku dali smo neke osnovne savete o pripremi elemenata za Unity, imajući na umu optimalnu performansu. Ovi saveti su uopšteni i primenjuju se za skoro sve tipove elemenata u skoro svim slučajevima, uključujući i našu igru *Dead Keys*. Sada ćemo da se fokusiramo na kreiranje našeg projekta - fp igre sa zombijima *Dead Keys*. Ova igra se oslanja na mnogo elemenata, od mrežica i tekstura, do animacije i zvuka. Ovde ćemo da importujemo i konfiguriramo mnogo osnovnih elemenata, razmatrajući probleme optimizacije i teme koje se odnose na elemente. Ne treba da importujemo sve elemente odmah; često ćemo importovati elemente kasnije u toku razvoja, integrišući ih u postojeću biblioteku elemenata. U ovom odeljku se pretpostavlja da smo već kreirali novi Unity projekat. Sada možemo da počnemo rad.

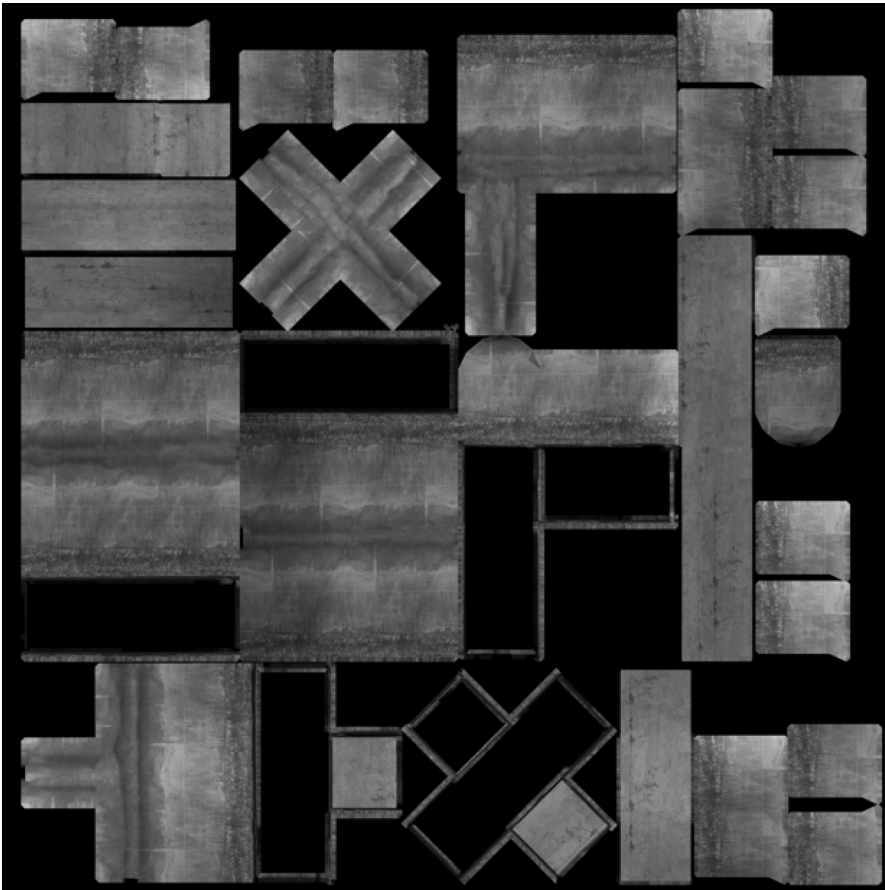
Na početku ćemo kreirati osnovnu strukturu direktorijuma na panelu Project tako da sadrži sve importovane elemente na sistematičan i organizovan način. Nazivi koje smo upotreбили u ovom primeru su jasni i opcioni. Imenovani direktorijumi su animation, audio, audio-mixers, Materials, meshes, music, prefabs, Resources, scenes, scripts i textures. Slobodno dodajte druge direktorijume ili promenite nazive, ako želite.



Organizovanje direktorijuma Project

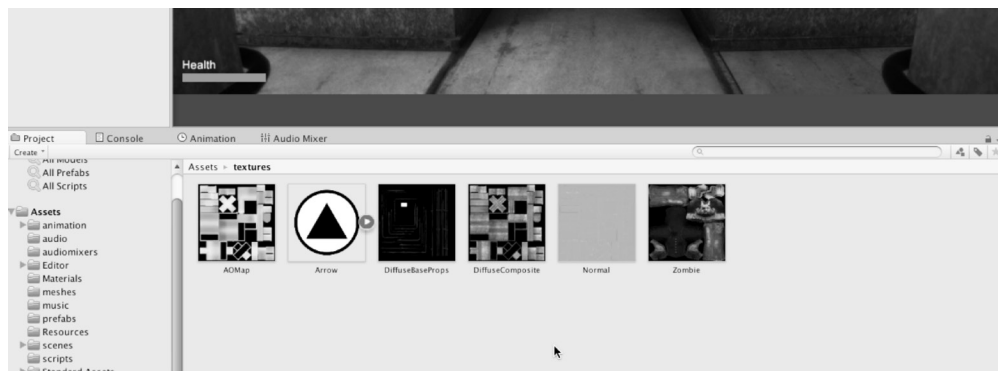
Importovanje tekstura

Direktorijum textures će sadržati sve teksture koje će biti upotrebljene u projektu. Najvažnije je da on uključuje teksture za NPC zombi karaktere (šake, ruke, noge i tako dalje) i modularni set okruženja. U igri *Dead Keys* okruženje će biti industrijski enterijer, prepun tamnih i sumornih hodnika i preseka. Ovo okruženje će stvarno biti sastavljeno od više manjih modularnih delova (kao što su preseki uglova i pravi preseki), koji su spojeni i koji su upotrebljeni više puta, kao što su gradivni blokovi za formiranje većih kompleksa okruženja. Svaki od delova u modularnom setu mapira UV prostor u istu teksturu (Texture Atlas), što znači da je celo okruženje, u stvari, u potpunosti mapirano pomoću jedne teksture. Pogledajte sada teksturu:



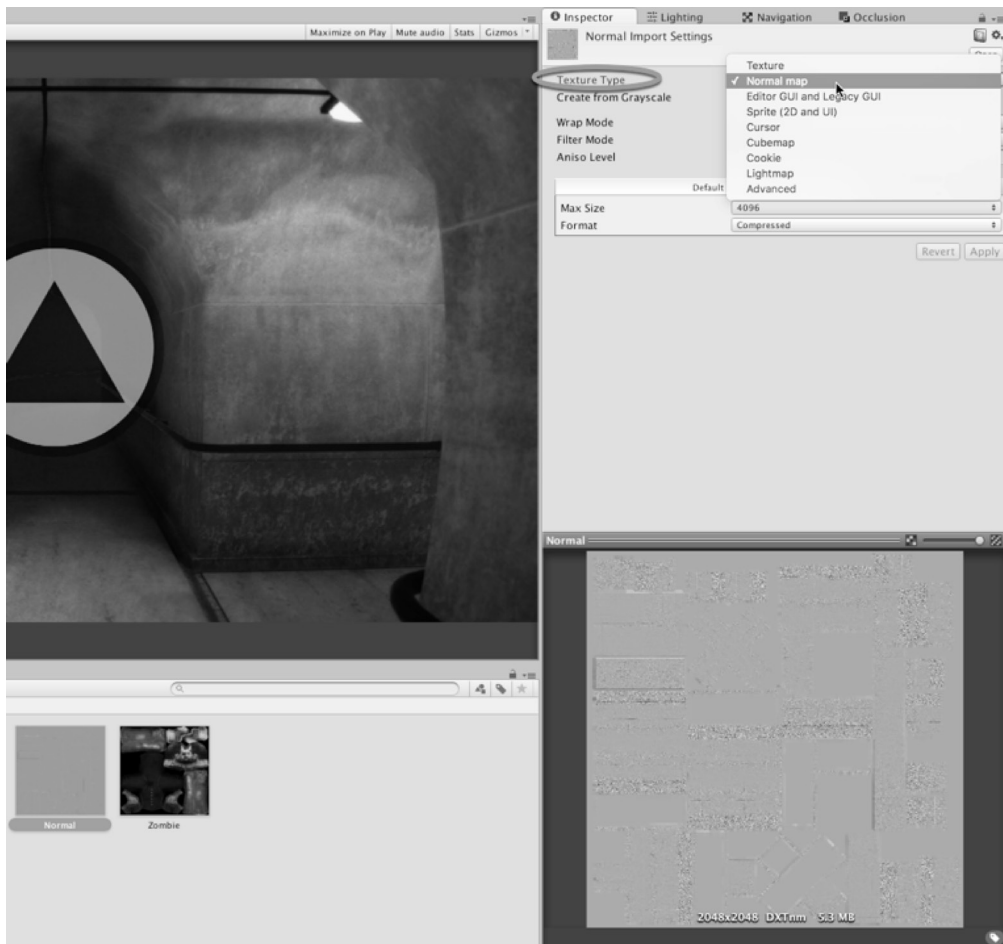
Atlas tekstura okruženja

Sve teksture za projekat su uključene u fajlove za ovu knjigu u direktorijumu `ProjectAssets/Textures`. Ovi fajlovi bi trebalo da budu importovani u Unity projekat jednostavnim prevlačenjem i otpuštanjem na panel **Project**. Pomoću ovog metoda možete da importujete više fajlova tekstura kao jednu grupu na sledeći način:



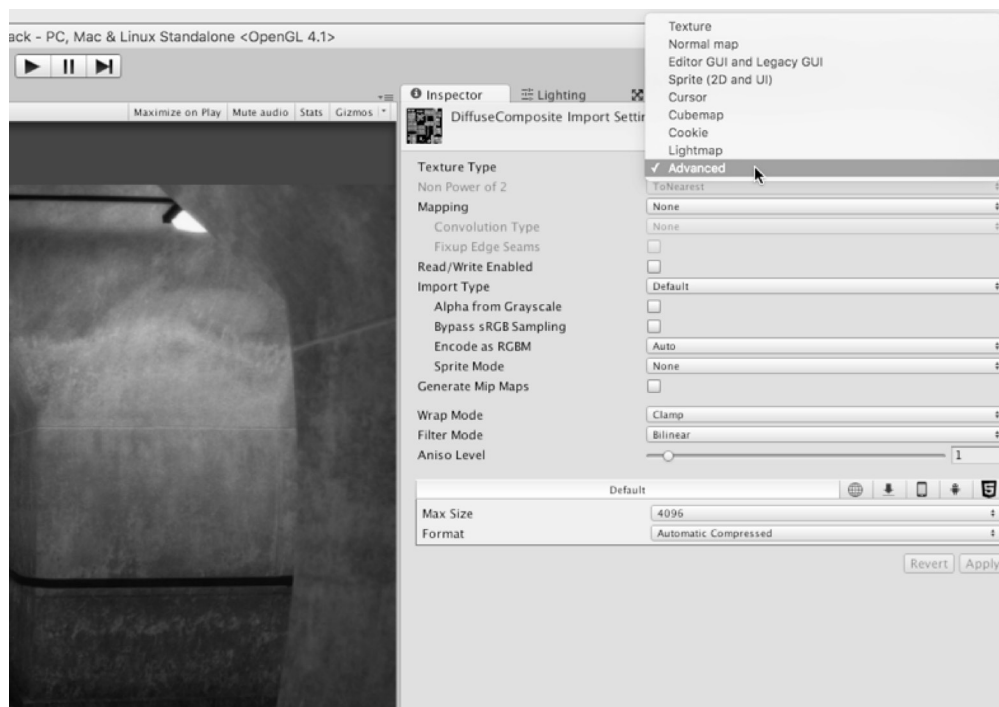
Importovanje tekstura u projekat

Prema standardnom podešavanju, Unity nepravilno konfiguriše Normal Map teksture kao regularne teksture. Ne razlikuje tip teksture na osnovu sadržaja slike. Prema tome, nakon importovanja Normal Mapa, treba da konfigurišemo svaku teksturu. Selektovaćemo **Normal map** na panelu **Project** i izabraćemo opciju **Normal Map** iz padajućeg menija Texture Type u objektu Inspector. Nakon toga ćemo kliknuti na **Apply** da bismo prihvatili promene:



Importovanje i konfigurisanje Normal Mapa

Pošto svaka mrežica u modularnom okruženju podešava mape u isti prostor teksture (uglovi, pravi preseki, okreti i tako dalje), treba da izvršimo neka manja podešavanja u podešavanjima Atlas Texture da bismo dobili najbolje rezultate. Prvo ćemo selektovati **Atlas Texture** na panelu **Project** (DiffuseComposite.png) i promenimo podešavanje **Texture Type** na **Advanced** iz objekta **Inspector**, što će nam omogućiti veću kontrolu nad podešavanjima teksture:



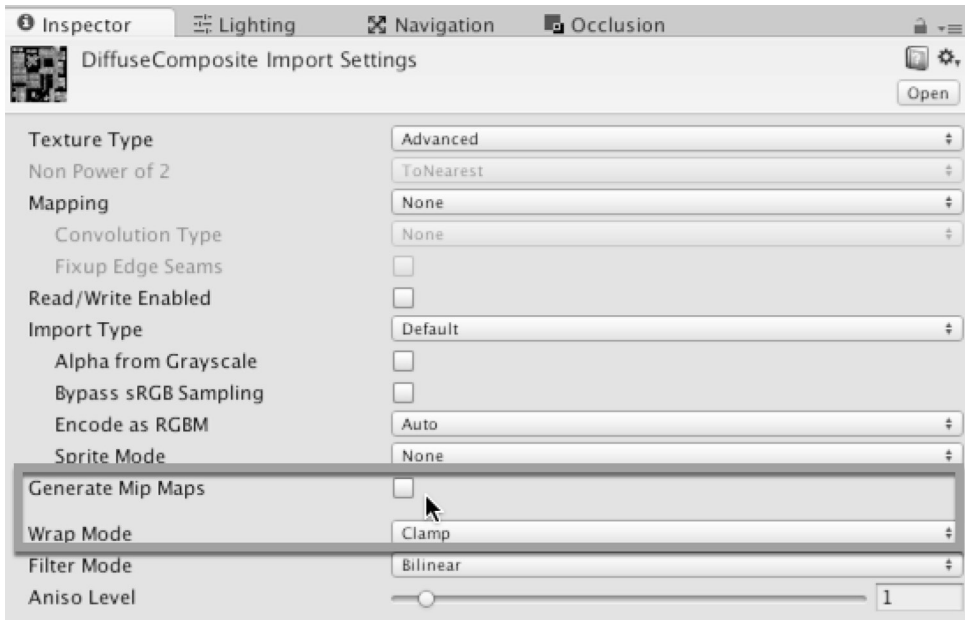
Pristupanje naprednim svojstvima teksture

Da bismo minimizirali šavove, prekide i artefakte u teksturi okruženja kad god se dva dela mrežice susreću na sceni, promenimo **Wrap Mode** teksture sa **Repeat** na **Clamp**. Režim Clamp obezbeđuje da pikseli ivice UV ostrva budu kontinualno skupljeni preko mrežice, umesto da se ponavljaju, ako je to potrebno. Ovo je korisna tehnika za smanjivanje šavova ili artefakta za mrežice koje se mapiraju u Texture Atlas.

Osim toga, uklonićemo oznaku potvrde iz opcije **Generate Mip Maps**. Kada je aktivirana, ova korisna optimizacija prikazuje progresivno teksture manjeg kvaliteta za mrežicu dok se ona pomera dalje od kamere. Ovo pomaže da se optimizuje performansa renderovanja pri pokretanju. Međutim, za Texture Atlas ovo može da bude problematično, jer promena veličine teksture u Unityju izaziva artefakte i šavove na ivicama UV ostrva gde god se susreću moduli mrežice, a to dovodi do mešanja piksela i distorzije u teksturama.

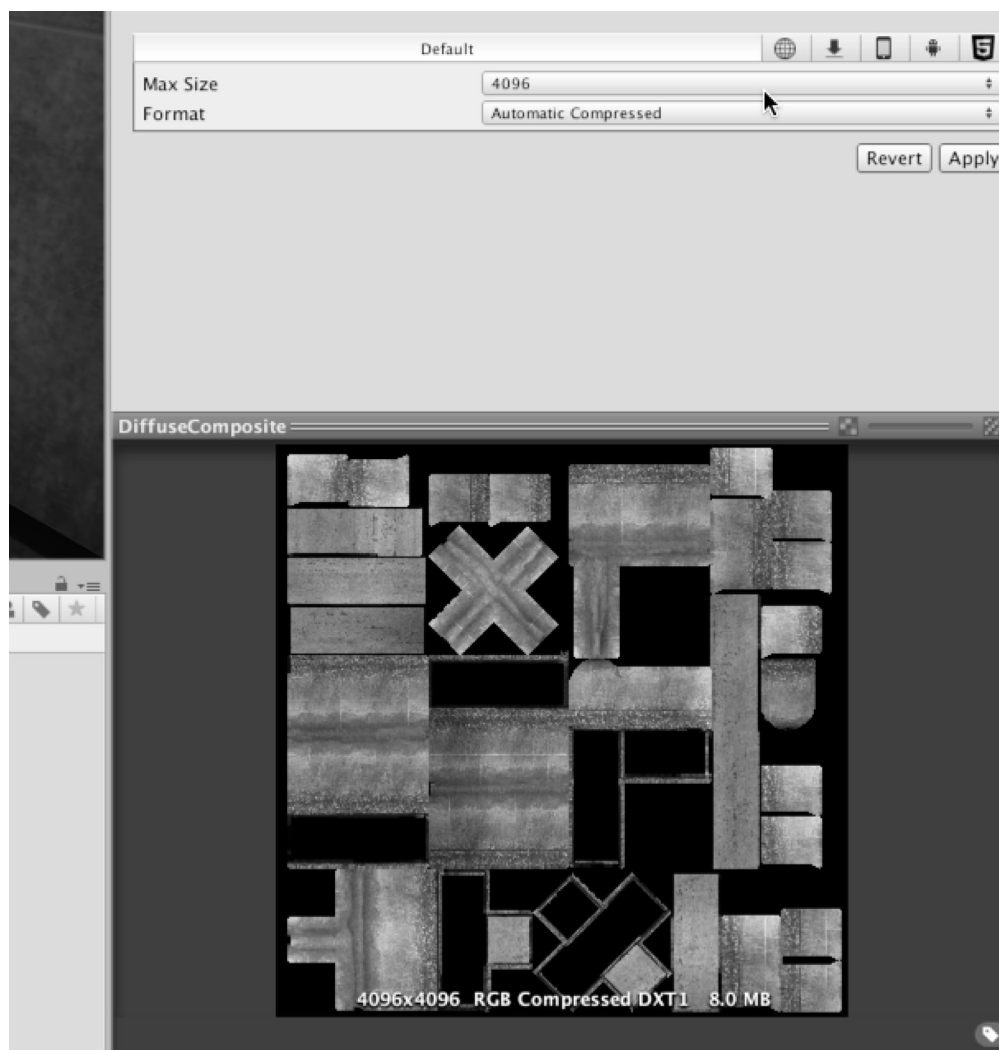


Ako želite da upotrebite Mip Maps za Atlas Textures bez rizika pojave artefakta, možete unapred da generišete sopstvene nivoe Mip Mapa - da kreirate teksture manjeg kvaliteta koje su kalibrisane specifično za upotrebu modularnih mrežica. To možda zahteva ručno i ponovno testiranje dok ne dobijete teksturu koja vam odgovara. Možete da generišete sopstvene Mip Map nivoe za Unity, tako što ćete eksportovati DDS teksturu iz Photoshopa. DDS format omogućava da specifikujete prilagođene Mip Map nivoe direktno u fajlu slike. Možete da preuzmete DDS plugin za Photoshop sa adrese <https://developer.nvidia.com/nvidia-texture-tools-adobe-photoshop>.



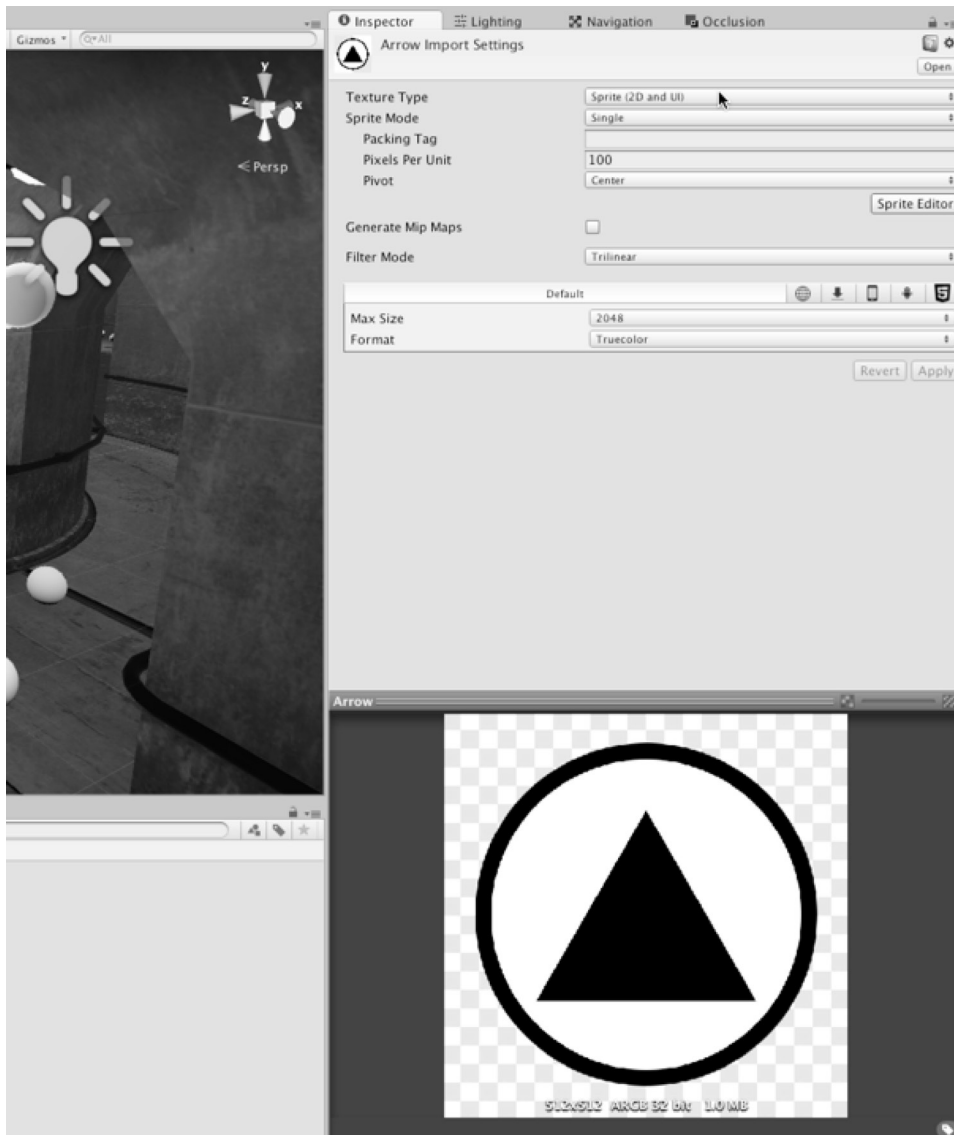
Optimizacija Atlas tekstura

Na kraju, specifikujte maksimalnu validnu power-2 veličinu za Atlas Texture, a to je **4096**. Izabrani format može da bude **Automatic Compressed**, koji će izabrati najbolji dostupan metod kompresije za desktop platformu; nakon toga, kliknite na Apply:



Primena promena u Atlas Texture teksturi

U ovom poglavlju ćemo izostaviti probleme u vezi sa korisničkim interfejsom. Međutim, sve GUI teksture treba da budu importovane kao tip teksture **Sprite (2D and UI)**, sa uključenom opcijom Mip Maps. Za UI teksture nije potrebno pratiti pravilo veličine power-2 (odnosno, veličine piksela 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 i tako dalje).



Importovanje UI tekstura

Importovanje mrežica

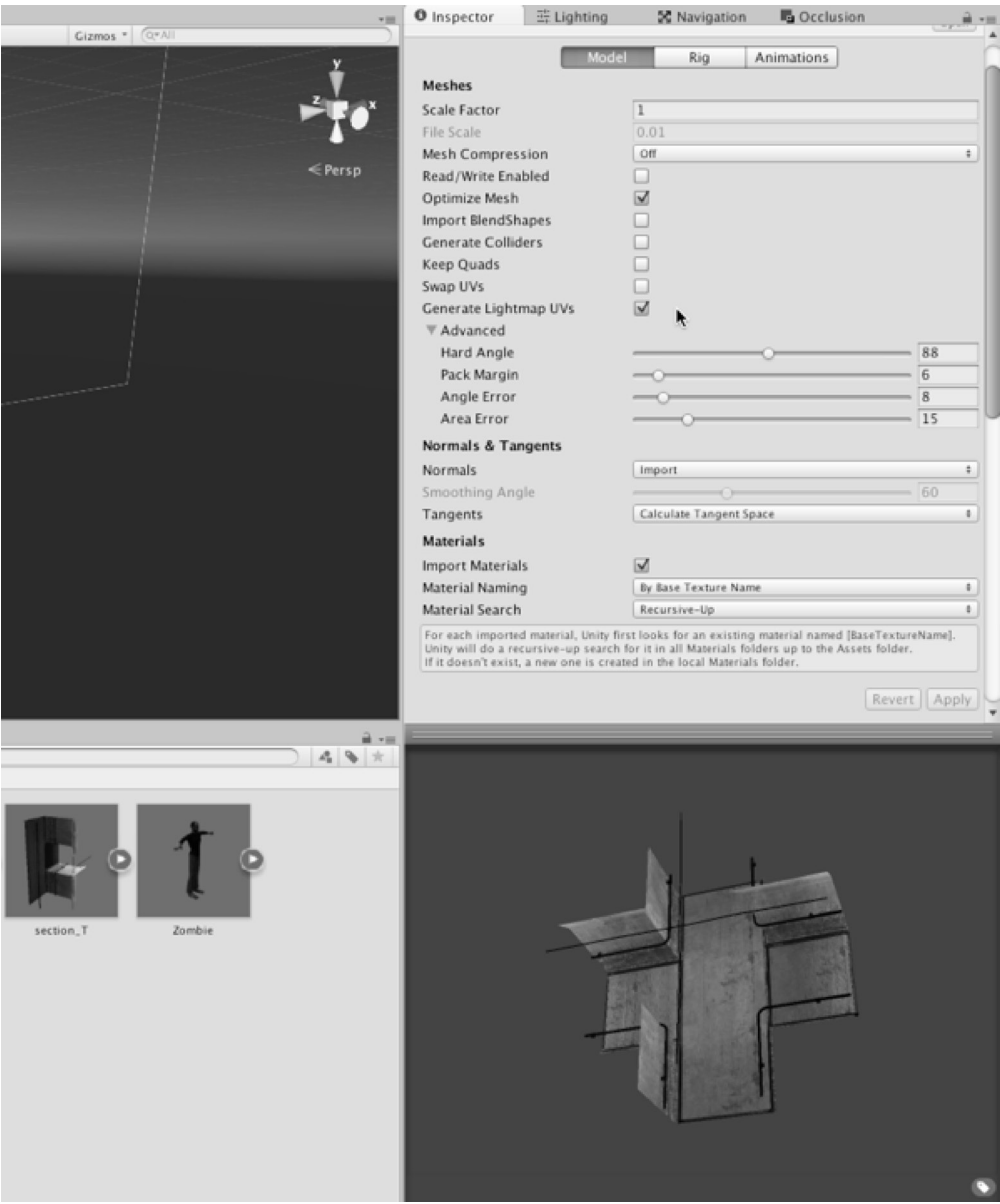
Idealno bi bilo da importujemo teksture pre mrežica, kao što smo i uradili u ovom primeru, zato što prilikom importovanja mrežice Unity automatski kreira materijale i pretražuje projekat za sve povezane teksture. Kada pronađe odgovarajuće teksture, dodeljuje ih materijalima pre prikaza rezultata na mrežici, čak i u umanjenom prikazu panela **Project**. Kada smo spremni da importujemo mrežice, treba samo da ih prevučemo i otpustimo na panel **Project** u namenski direktorijum meshes. Unity importuje na taj način sve mrežice kao jednu grupu. Ovaj projekat se oslanja na mrežice i animirane mrežice karaktera za NPC zombije i statične mrežice okruženja za modularno okruženje – kao i na prop mrežice i bilo koje druge mrežice koje bismo želeli da uključimo u naš kreativan rad. Ti fajlovi (osim sopstvenih mrežica) su uključeni u fajlove za ovu knjigu.



Importovanje mrežica (mrežice okruženja i karaktera)

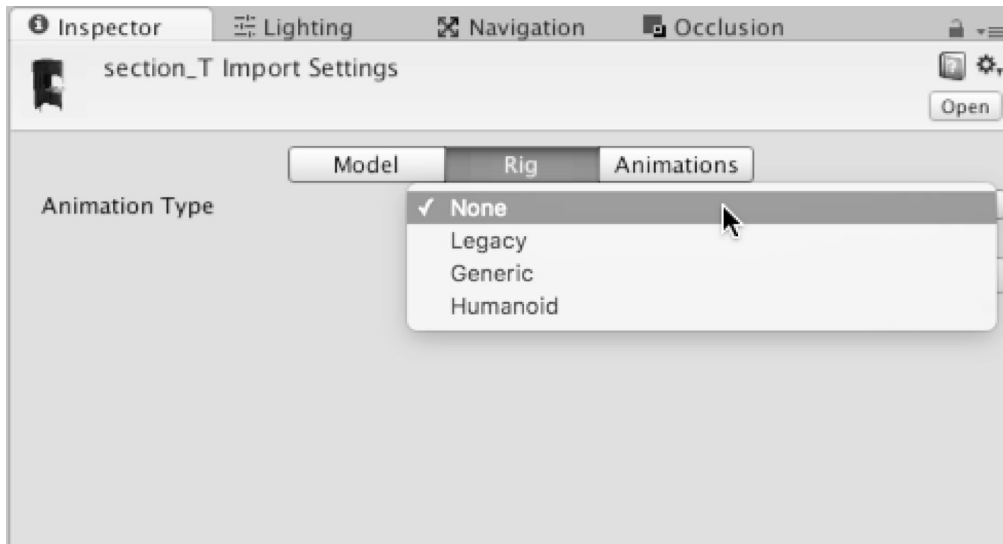
Sada ćemo da konfigurišemo modularne mreže okruženja. Selektovaćemo sve mrežice za okruženje, uključujući `section_Corner`, `section_Cross`, `section_Curve`, `section_End`, `section_Straight` i `section_T`. Kada su selektovane mrežice, podesićemo sledeća podešavanja:

- Podesićemo **Scale Factor** za mrežicu na 1, kreirajući odnos 1:1 između modela, kao da je kreiran u softveru za modelovanje, i tako će model biti prikazan u Unityju.
- Isključićemo opciju **Import BlendShapes**. Mrežice okruženja ne sadrže složene oblike za importovanje i možemo da poboljšamo proces importovanja i ponovnog importovanja isključivanjem nepotrebnih opcija.
- Isključićemo opciju **Generate Colliders**. U mnogim slučajevima ovo podešavanje će biti uključeno. Međutim, Dead Keys je fps igra sa fiksnom, AI kontrolisanom kamerom, umesto sa slobodno lutajućom kamerom. To igrača sprečava da prolazi kroz zidove ili kroz podove.
- Uključićemo opciju **Generate Lightmap UVs**. Njeno uključivanje generiše drugi UV kanal. Unity automatski odvija mrežice i garantuje da neće biti preklapanja UV ostrva. Možemo dalje da podesimo generisanje UV mape osvetljenja, koristeći podešavanja **Hard Angle**, **Pack Margin**, **Angle Error** i **Area Error**. Međutim, u većini situacija standardna podešavanja funkcionišu bolje. Podešavanje **Pack Margin** može (i verovatno bi trebalo) da bude povećano ako je rezolucija mape osvetljenja niska, kao što ćete videti u sledećem poglavlju. Podešavanja ugla i greške bi ponekad trebalo da budu povećana ili smanjena da bi mape osvetljenja bile bolje prilagođene za organske i zakrivljene površine.



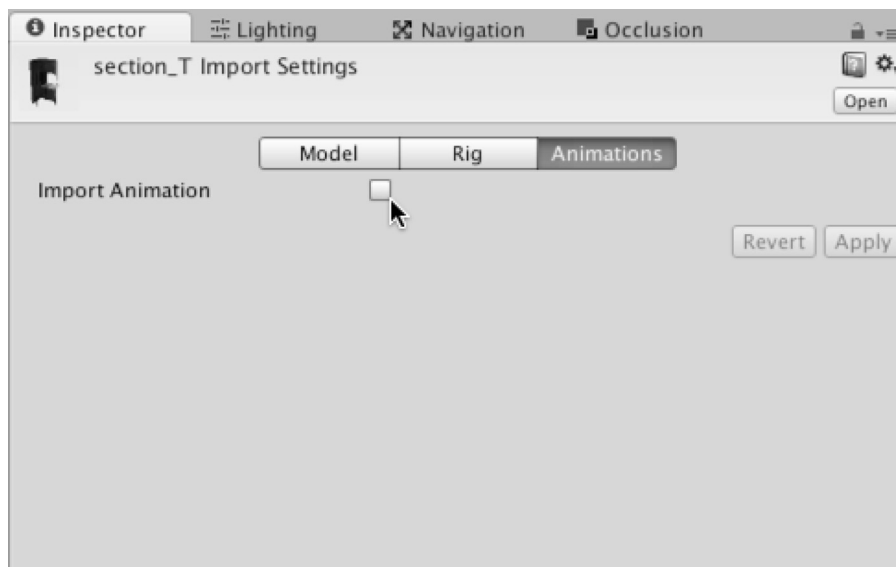
Konfigurisanje mrežica okruženja

Osim što možemo da konfigurišemo primarna svojstva mrežice, kao što smo videli, takođe možemo da se prebacimo na kartice **Rig** i **Animation**. Na kartici **Rig** specificovaćemo **None** za polje **Animation Type**, jer mrežice ne sadrže podatke animacije.



Podešavanje Rig tipa za mrežice okruženja

Zatim ćemo kliknuti na karticu Animation. **Uklonićemo** oznaku potvrde za opciju **Import Animation**. Mrežice okruženja nemaju animacije za importovanje. Zatim ćemo kliknuti na **Apply**:



Isključivanje opcije Import Animation

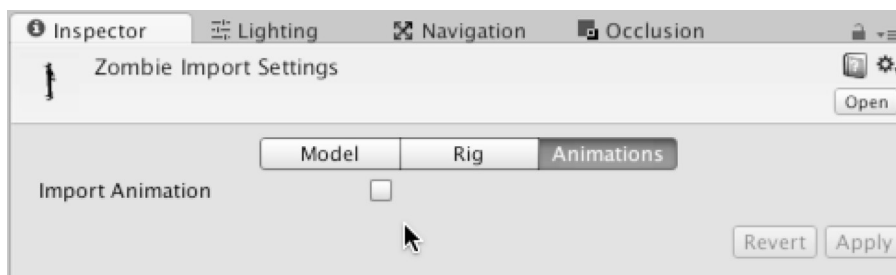
Naravno, u igri *Dead Keys* je reč o izvršavanju vežbi kucanja za uništavanje zombija. Karakter zombija za naš projekat je zasnovan na karakteru zombija javnog domena, koji je dostupan na sajtu *Blend Swap* na adresi <http://www.blendswap.com/blends/view/76443>. Ovaj karakter je kreiran i konfigurisan u Blenderu radi lakšeg importovanja u Unity. Hajde da konfigurišemo ovaj karakter. Selektovaćemo mrežicu Zombie u panelu Project; a zatim ćemo u objektu Inspector podesiti sledeća podešavanja:

- Opciju **Mesh Scale Factor** podesićemo na 1 da bismo zadržali originalnu veličinu.
- Uključićemo opciju **Import BlendShapes** da bismo omogućili animaciju temena.
- Isključićemo opciju **Generate Colliders**, jer detekcija sudara nije potrebna.
- Na panelu prikaza uključićemo opciju **Swap UVs** ako tekstura ne izgleda dobro na modelu zombija. Ako objekat ima dva ili više UV kanala (a ponekad ima), Unity će povremeno selektovati pogrešan kanal.



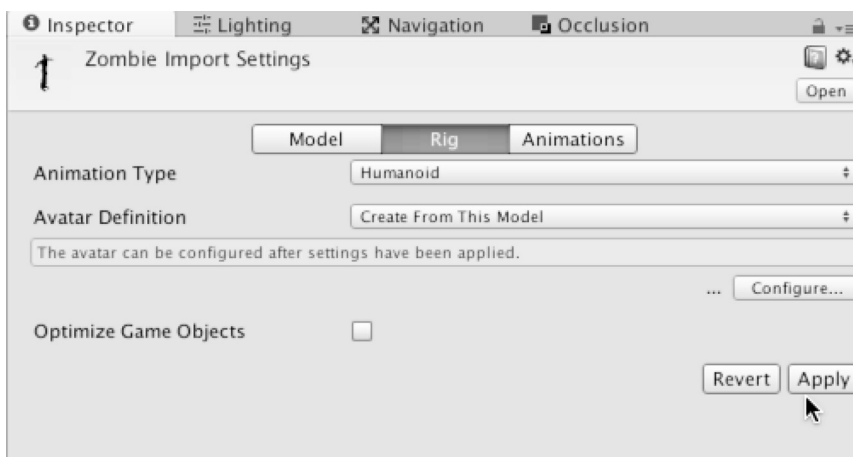
Konfigurisanje zombi NPC-a

Prebacićemo se na karticu Animations i isključićemo opciju **Import Animation**. Mrežica karaktera bi trebalo da bude animirana, izvršavajući akcije, kao što su animacije hoda i napada. Međutim, sam fajl mrežice karaktera ne sadrži podatke animacije. Sve animacije karaktera će biti primenjene na mrežici iz drugih fajlova.



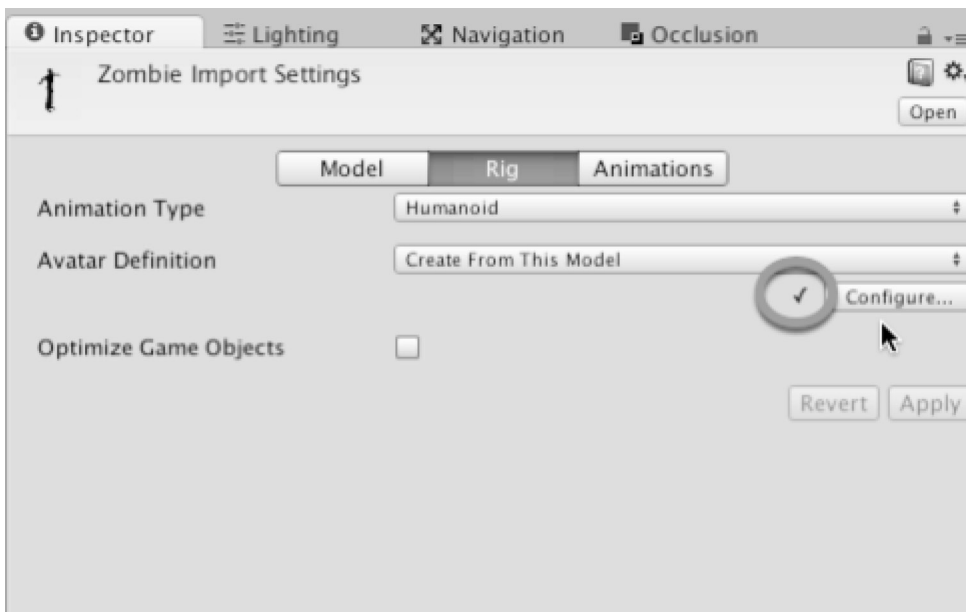
Isključivanje opcije Import Animation za NPC zombijao

Sada ćemo da konfiguriramo opremu karaktera za Mecanim. To je optimizovanje osnovnog skeleta da bismo omogućili da model bude animiran. Da bismo to uradili, selektovaćemo karticu **Rig** iz objekta **Inspector**. Za Animation Type izabraćemo **Humanoid**, a za Avatar Definition opciju **Create From This Model**. Tip animacije **Humanoid** daje instrukcije Unityju da vidi mrežicu kao standardni dvonožni ljudski lik – karakter koji ima glavu, torzo, dve ruke i dve noge. Ova generička struktura (definisana u avataru) je mapirana u kosti mrežice i omogućava podešavanje **Animation Retargeting**. Animation Retargeting je mogućnost upotrebe i ponovne upotrebe animacija karaktera iz drugih fajlova i drugih modela u svakom ljudskom liku.



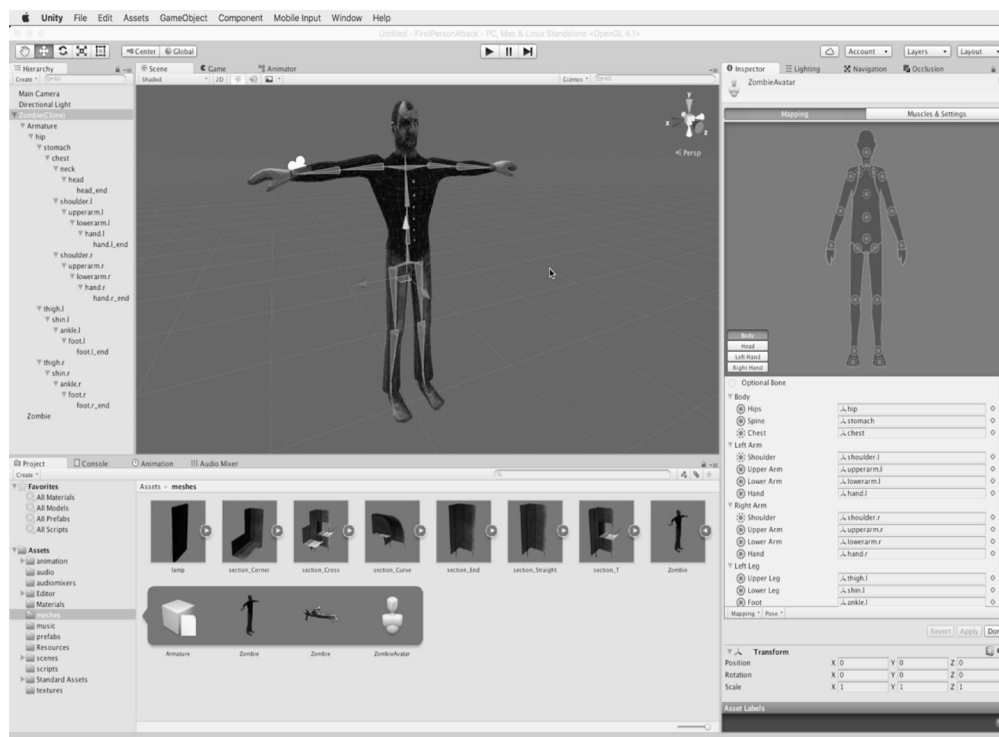
Konfigurisanje opreme zombija

Nakon što kliknemo na dugme Apply za karakter zombija, biće prikazana ikonica oznake sa potvrdom pored dugmeta **Configure...**. Za neke mrežice karaktera biće, umesto toga, prikazana ikonica X. Oznaka potvrde označava da je Unity skenirao sve kosti u mrežici i uspešno identifikovao ljudski lik, koji može lako da bude mapiran u avatar. Ikonica X označava problem koji može biti mali ili veliki. Mali problem je kada je karakter ljudskog lika importovan, ali se razlikuje na suptilan i važan način od onoga što Unity očekuje. Ovaj scenario se često ispravlja ručno u Unityju pomoću okvira za dijalog **Rig Configuration Window** (koji je dostupan ako kliknemo na **Configure...**). Nasuprot tome, problem može da bude i veliki; na primer, importovana mrežica možda uopšte nije ljudski lik ili se drastično razlikuje od svega što se očekuje, pa moraju da budu izvršene radikalne promene na karakteru unutar softvera za kreiranje sadržaja.



Oprema karaktera je uspešno konfigurisana.

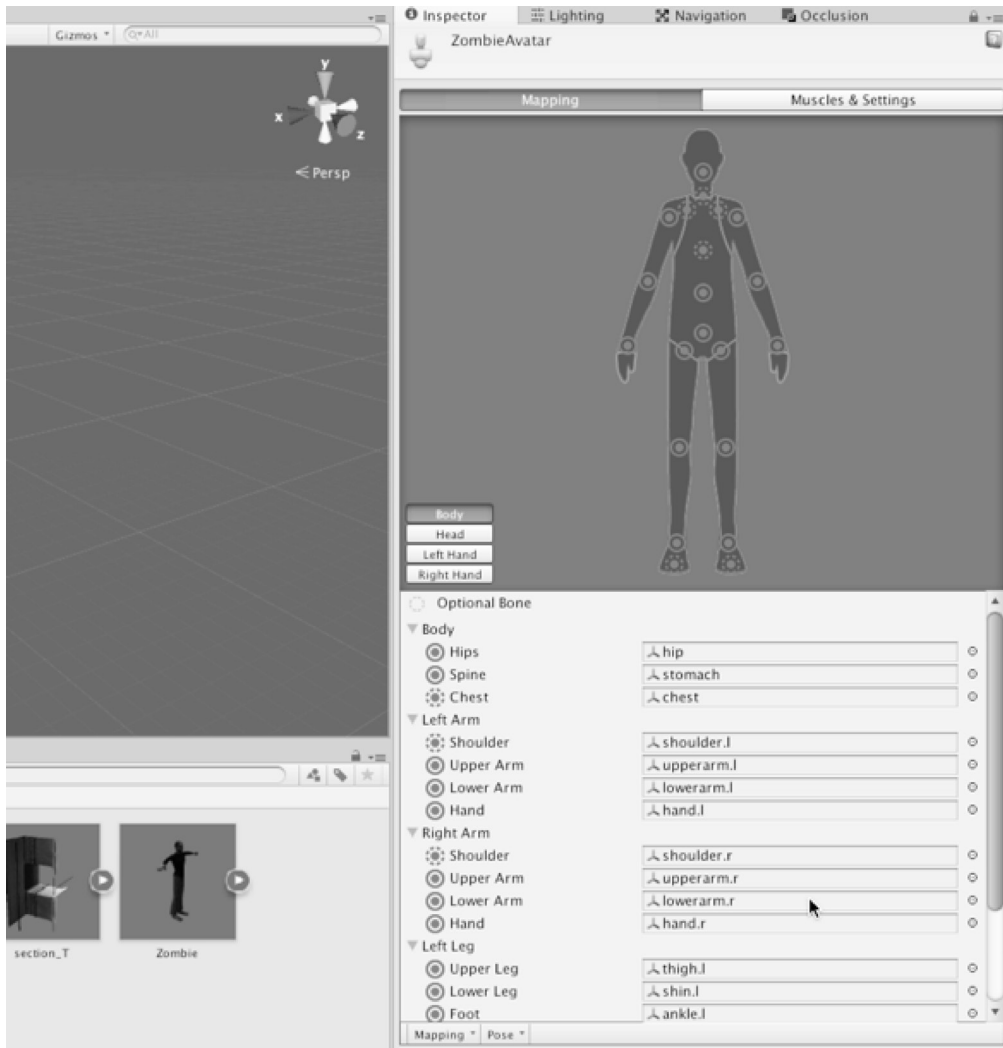
Čak i kada je karakter uspešno importovan, treba da ga testiramo unutar editora **Rig Configuration Editor**. Ovim testom se *proverava razum* i utvrđuje da li oprema funkcioniše onako kako bi trebalo. Da bismo ga izvršili, kliknućemo na dugme **Configure...** na kartici **Rig** u objektu **Inspector**; biće prikazan prozor Rig Configuration Editor:



Upotreba prozora Rig Configuration Editor za ispitivanje, testiranje i ispravku mapiranja skeleta avatara

Iz prozora Rig Configuration Editor možemo da vidimo kako se importovani kostur mapira u definiciju avatara ljudskog lika. Kostur istaknut zelenom bojom je već mapiran u element Avatar, kao što je prikazano u objektu **Inspector**. Odnosno, importovani kostur će postati zelen kada Unity, nakon analize, otkrije podudarnosti u avatara. Avatar je, jednostavno, mapa ili grafikon koji definiše Unity - konkretno, kolekcija unapred definisanih kostura. Cilj je da prozor Rig Configuration Editor jednostavno mapira kostur iz mrežice u avatar, omogućavajući da mrežica bude animirana bilo kojom vrstom animacije ljudskog lika.

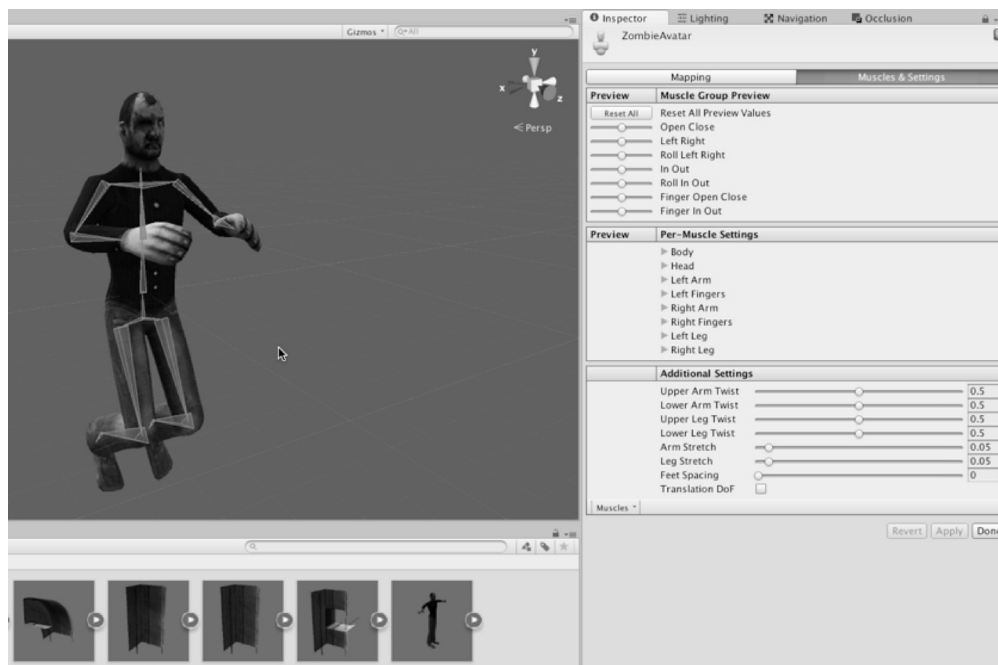
Za karakter zombija sve kosti će biti uspešno automatski mapirane u avatar. Možemo da promenimo mapiranje jednostavnim prevlačenjem i otpuštanjem određene kosti iz panela **Hierarchy** u slotove kostura u objektu **Inspector**.



Definisanje mapiranja avatara

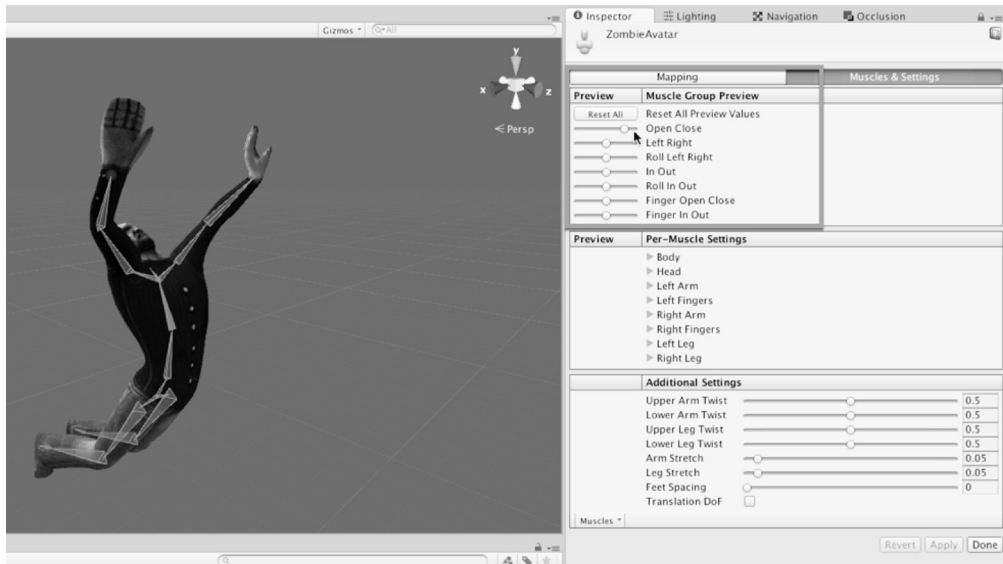
POGLAVLJE 1 Priprema i konfigurisanje elemenata postavke

Sada ćemo izvršiti test opterećenja mrežice karaktera, proveravajući kostur i mapiranje avatara da bismo se uverili da se karakter deformiše onako kako očekujemo. Da bismo izvršili ovaj test, kliknućemo na karticu **Muscles & Settings** u objektu **Inspector**. Kada to uradimo, poza karaktera se odmah menja unutar prozora prikaza, što znači da je on spreman za testiranje.



Testiranje mapiranja kostura

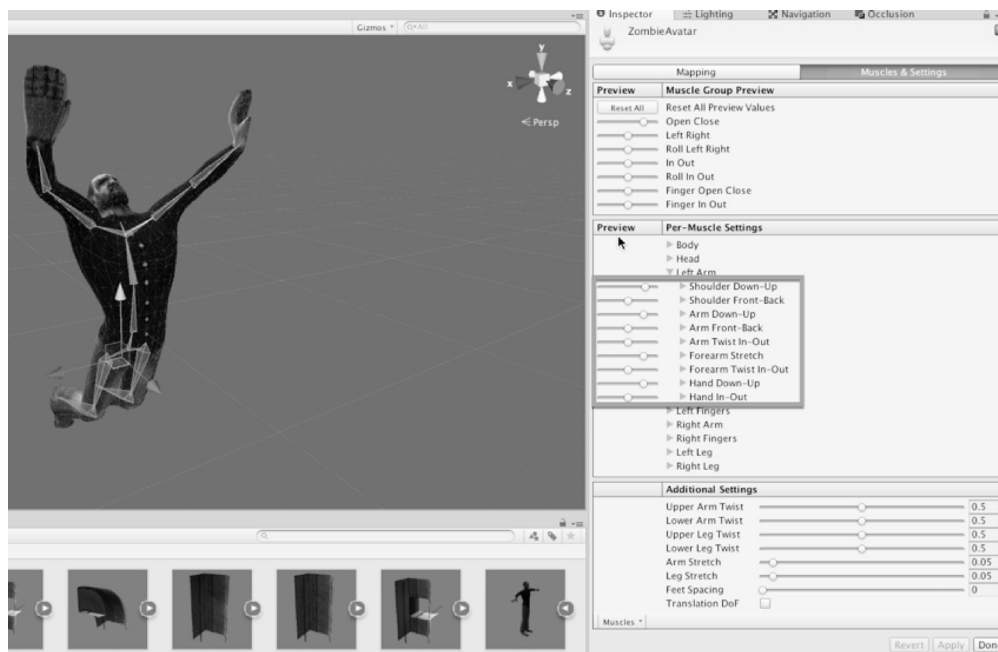
Upotrebićemo klizače poze karaktera u objektu **Inspector** da bismo pomerili karakter u ekstremne poze, pregledajući rezultate u prozoru prikaza. Ideja je da prikazemo kako se karakter deformiše i odgovara na ekstremne promene. Razlog potrebe takvog testiranja je što se, iako dvonožni ljudski likovi dele zajedničku strukturu skeleta, veoma razlikuju po tipu tela i visini – neki su niski i mali, a neki visoki i veliki.



Testiranje ekstremnih poza

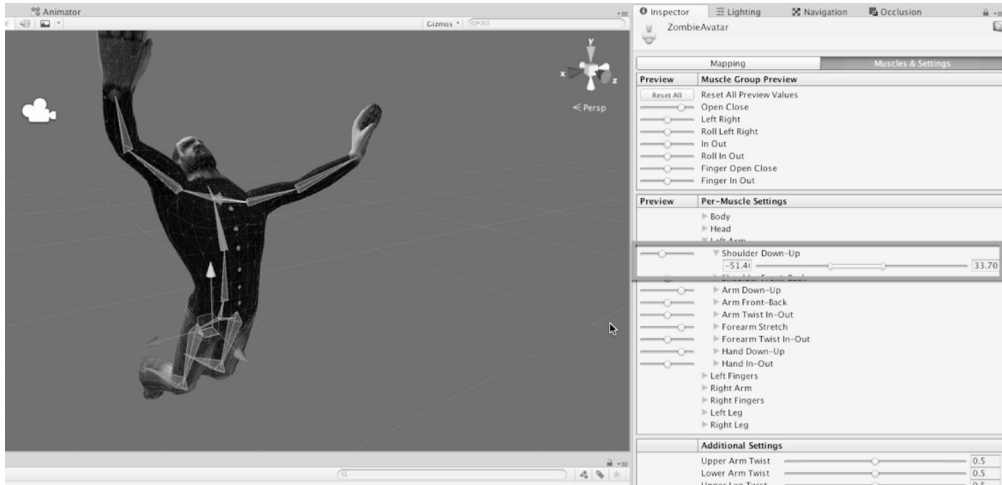
POGLAVLJE 1 Priprema i konfigurisanje elemenata postavke

Ako mislite da se karakter lomi, preseca ili izobličuje u ekstremnim pozama, možete da konfigurirate ograničenje deformacije mrežice, specifikujući minimalni i maksimalni raspon. Da biste to uradili, prvo proširite grupu **Per-Muscle Settings** za udove ili kosture koji su problematični, kao što je prikazano na sledećoj slici.



Definisanje ekstremnih poza

Zatim, možete da prevučete i promenite veličinu klizača minimum i maximum da biste definisali manimalnu i maksimalnu deformaciju za određeni ud i za sve udove gde je to potrebno. Ova podešavanja ograničavaju pokret i rotaciju udova - da ne budu pomereni iza namenjene granice u toku animacije. Najbolji način da upotrebimo ovu alatku je da počnemo kreiranje karaktera u ekstremnoj pozi koja izaziva vidljiv prelom, a zatim da fino izvršimo podešavanje **Per-Muscle Settings**, dok mrežica ne bude ispravljena.



Ispravljanje preloma poze

POGLAVLJE 1 Priprema i konfigurisanje elemenata postavke

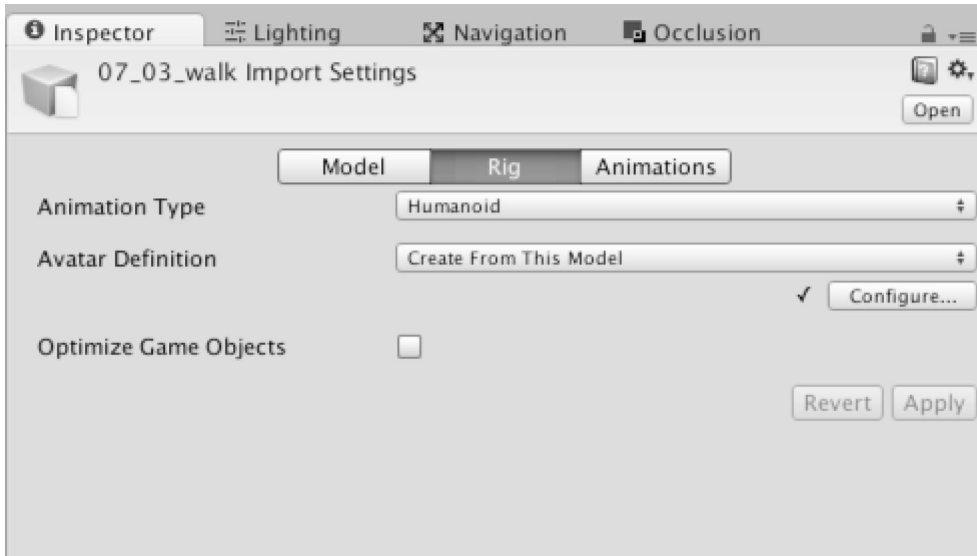
Kada izvršite promene za karakter i poze, ne zaboravite da kliknete na dugme **Apply** ili **Done** u objektu **Inspector**. Dugme **Done** jednostavno primenjuje promene, pa zatvara prozor Rig Configuration Editor.



Primena promena za karakter

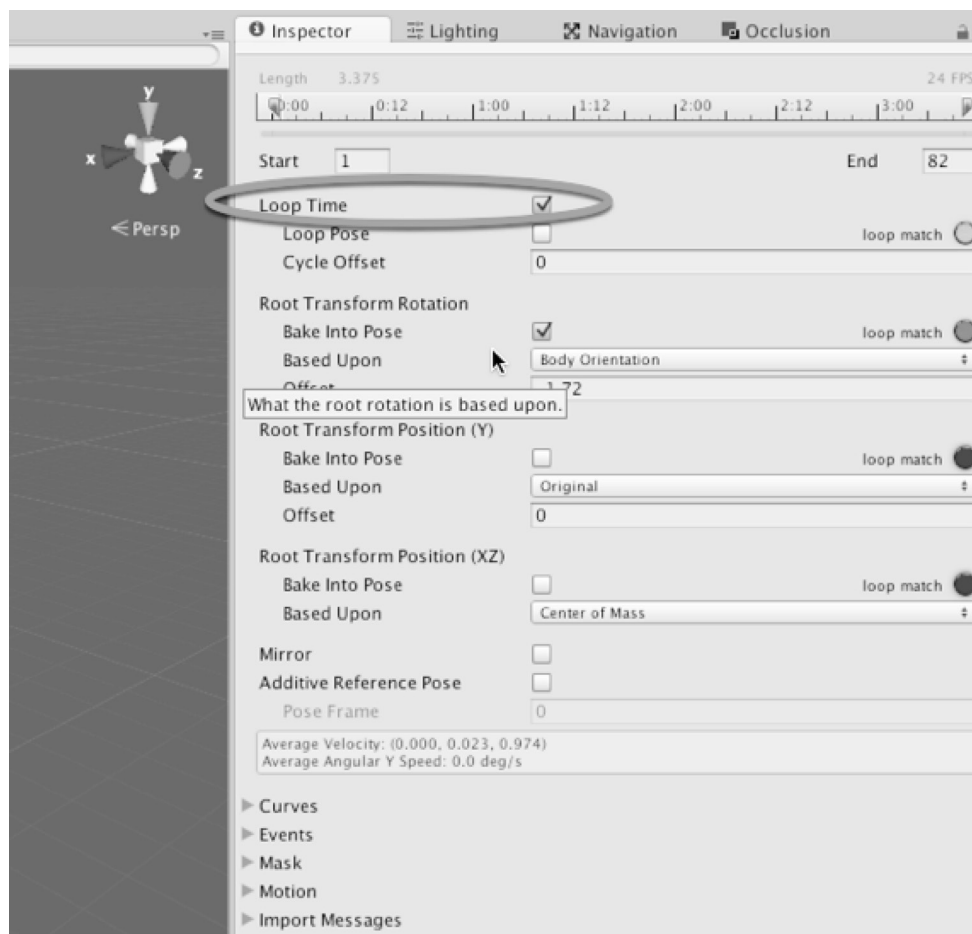
Importovanje animacija

Igra Dead Keys ima animacije karaktera za zombije – konkretno, za hod, borbu i stanje mirovanja. Ove animacije uključuju FBX fajlove koji mogu da budu importovani u direktorijum *Animations*. Same animacije nisu namenjene za zombije, ali Mecanimov Humanoid Retargeting omogućava da upotrebimo skoro bilo koju animaciju karaktera na bilo kom modelu ljudskog lika. Hajde da konfigurišemo animacije. Selektovaćemo svaku animaciju i kliknućemo na karticu Rig. Izabraćemo **Humanoid** za **Animation Type**, a opciju **Avatar Definition** ćemo ostaviti podešenu na **Create From This Model**.



Specifikovanje tipa animacije Humanoid za animacije

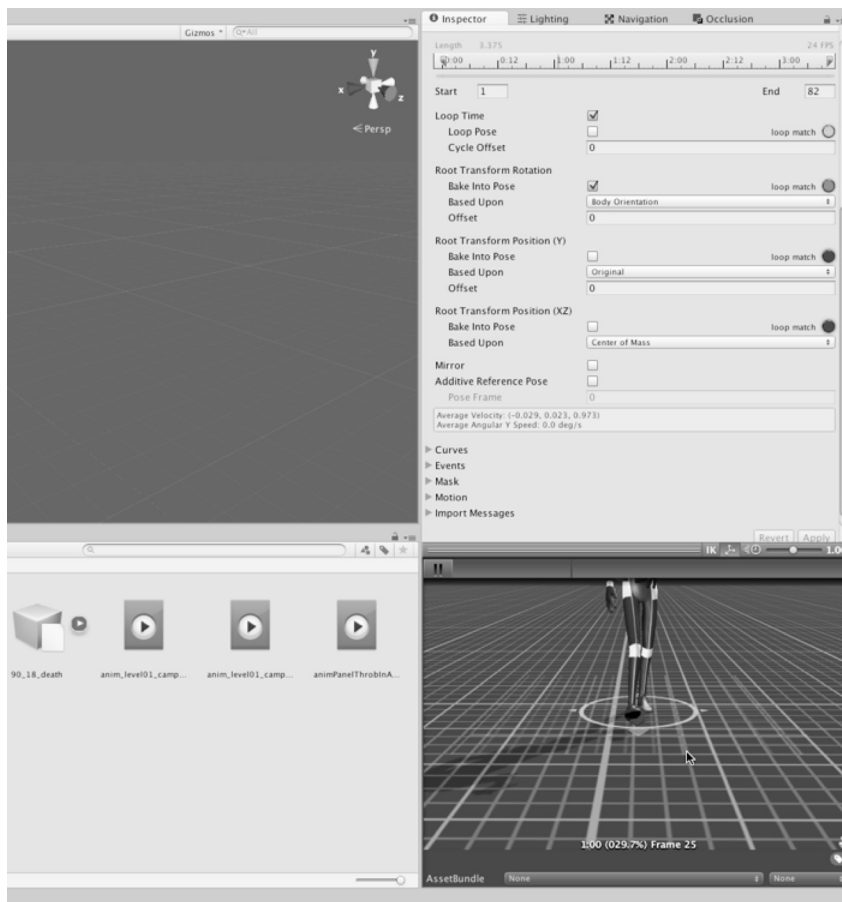
Kliknućemo na karticu Animations. Uključićemo polje za potvrđivanje **Loop Time** da bismo omogućili kruženje animacije za isečak. Zatim ćemo kliknuti na dugme **Apply**. Imaćemo dobar razlog da se vratimo na podešavanja animacije u narednim poglavljima radi daljeg podešavanja, kao što ćete videti.



Uključivanje opcije Loop Time za ponavljanje isečaka animacije

Sada ćemo da istražimo uobičajeni problem animacija hoda koje se ponavljaju i koje imaju kodiran osnovni pokret. Osnovni pokret se odnosi na transformaciju najvišeg nivoa, primenjenu na animirani model. Većina animacija zasnovanih na skeletu primenjuje skelete nižeg nivoa u hijerarhiji skeleta (kao što su ruke, noge i glava); ove animacije se uvek mere u odnosu na najviši matični skelet.

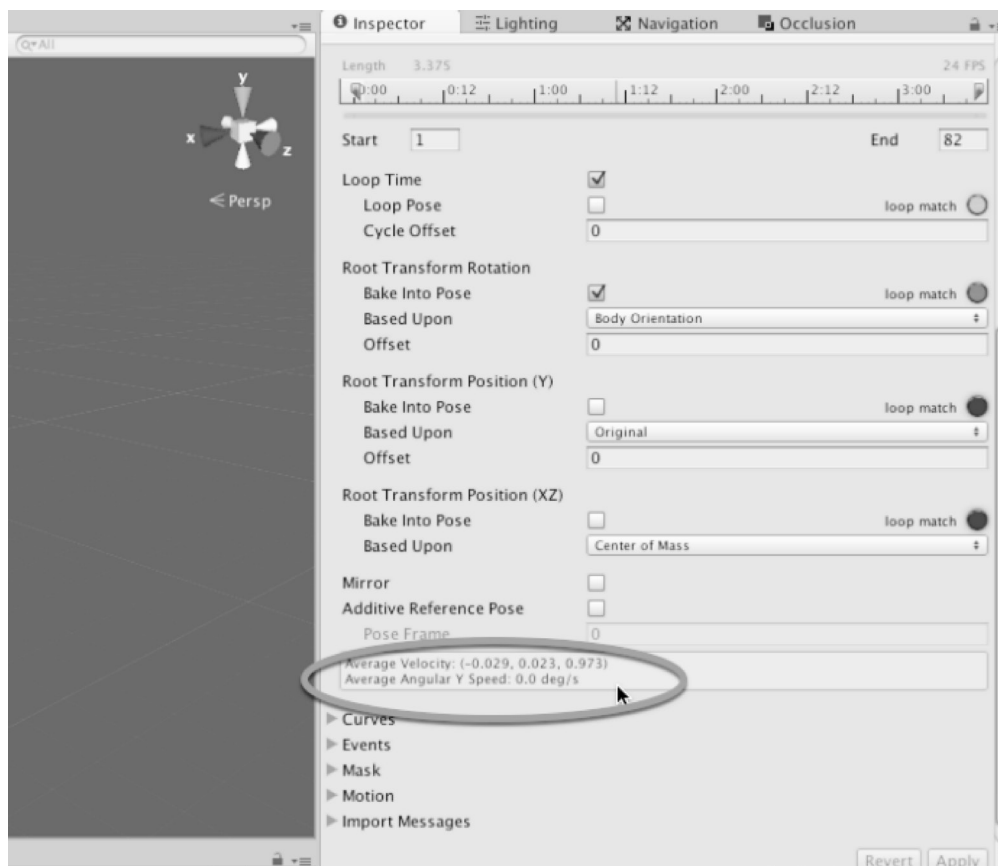
Međutim, kada je osnovni skelet animiran, to utiče na poziciju karaktera i orijentaciju u prostoru, što je poznato kao osnovni pokret. Problem koji se ponekad javlja u importovanim animacijama hoda koji se ponavlja je mala devijacija ili pomeranje od neutralne početne tačke u osnovnom pokretu. To izaziva da se mrežica vremenom pomeri sa svoje početne orijentacije, posebno kada je animacija reprodukovana u krug. Da bismo videli ovaj problem u akciji, selektovaćemo animaciju hoda za karakter zombija i u objektu **Inspector** pažljivo ćemo pregledati animaciju. Dok to radimo, poravnaćemo prikaz kamere u prozoru prikaza ispred karaktera ljudskog lika i videćemo kako njegov hod postepeno i polako odstupa od centralne linije na kojoj je započet. To prikazuje da će, vremenom, karakter nastaviti da odstupa od svoje pozicije. Ovaj problem neće se manifestovati samo u prozoru prikaza, već i u igri.



Prikaz problema ciklusa hoda

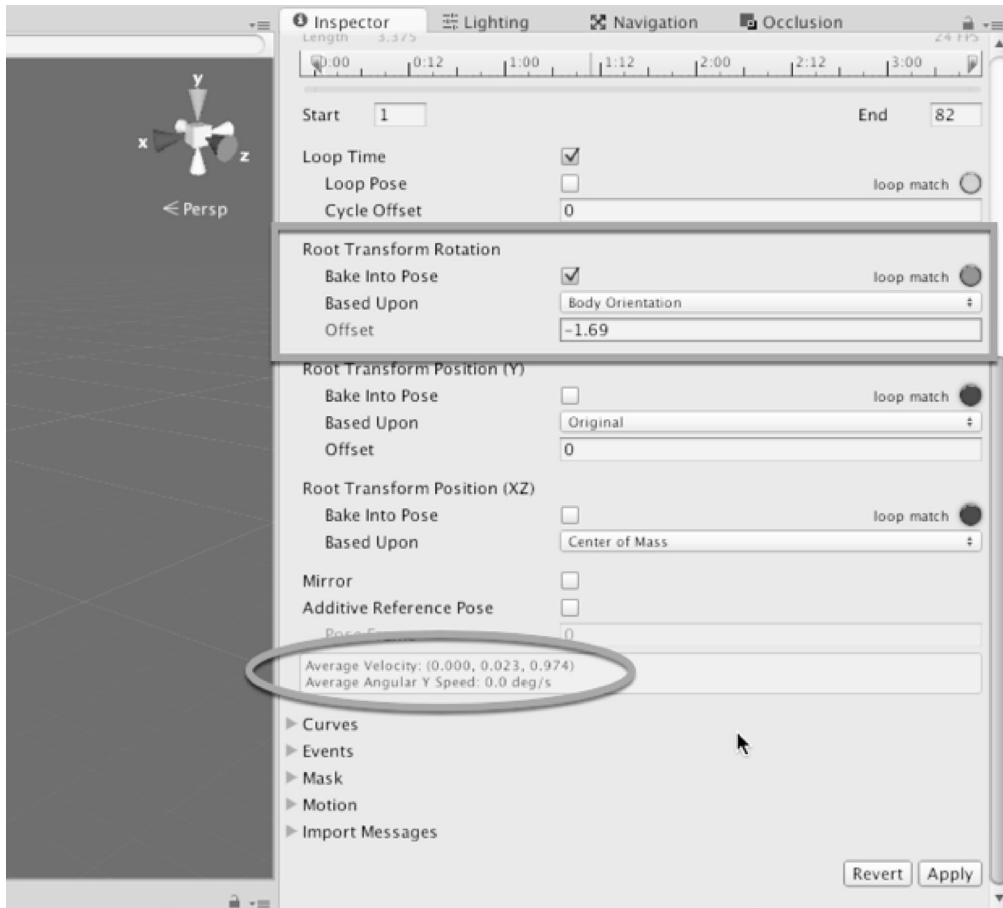
POGLAVLJE 1 Priprema i konfigurisanje elemenata postavke

Ovaj problem se pojavljuje kao rezultat netačnosti ciklusa hoda u osnovnom pokretu. Pregledom polja **Average Velocity** iz objekta **Inspector** videćete da polje **X** za pokret ima vrednost koja nije nula, što znači da se dešava odstupanje u mrežici na X osi. To objašnjava akumulativnu devijaciju u hodu kada se animacija ponavlja.



Ispitivanje problema osnovnog pokreta

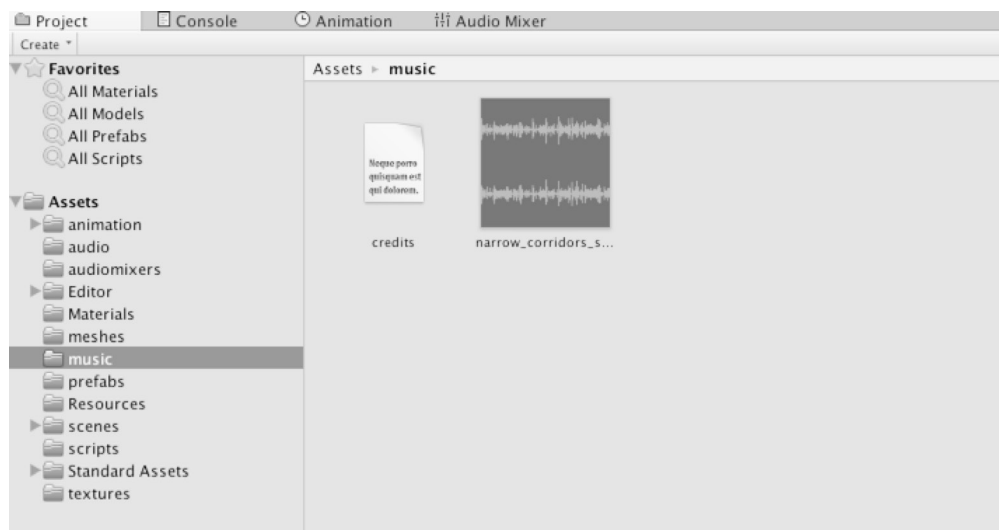
Da bismo otklonili ovaj problem, uključićemo polje **Bake Into Pose** za odeljak **Root Transform Rotation**, što će omogućiti da promenimo vrednost polja **Average Velocity**. Zatim ćemo da podesimo polje **Offset** da bismo nadoknadili vrednost vrednost polja **Average Velocity**. Ideja je da podešavamo polje **Offset** dok vrednost polja **Average Velocity** ne bude resetovana na 0, što ukazuje da nema odstupanja. Zatim ćemo kliknuti na **Apply**.



Ispravljanje osnovnog pokreta

Importovanje audio zapisa

Hajde da importujemo audio zapise za igru – konkretno, muzičku numeru. Fajl treba da prevučemo i otpustimo u direktorijum `music` (muzička numera `narrow_corridors_short.ogg` je uključena u fajlove za ovu knjigu). Muzika je važan audio element, koji u velikoj meri utiče na vreme učitavanja, posebno u mobilnim uređajima i naslednom hardveru. Muzičke numere često su duže od jednog minuta i kodiraju mnogo podataka. Shodno tome, obično je potrebno dodatno konfigurisanje za muzičke numere da bi bilo sprečeno opterećenje igre.



Importovanje audio fajlova



Idealno bi bilo da muzika bude u WAV formatu da bismo sprečili gubitaknu kompresiju kada prebacujemo igru na druge platforme. Ako nije moguće da upotrebite WAV, druga vredna alternativa je OGG format. Za više informacija o podešavanjima importovanja audio zapisa pogledajte Unity dokumentaciju na adresi <http://docs.unity3d.com/Manual/Audio-Files.html>.

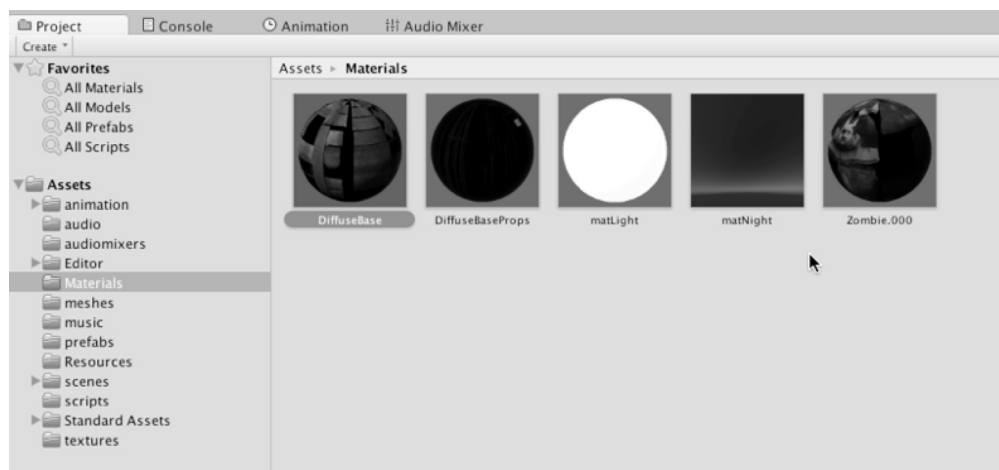
Sada ćemo selektovati importovanu muzičku numeru na panelu Project. Isključićemo opciju **Preload Audio Data**, a zatim ćemo promeniti **Load Type** na **Streaming**, što će optimizovati proces učitavanja muzike. To znači da će muzička numera biti učitana u segmentima u toku reprodukcije, umesto da bude potpuno u memoriji od početka nivo, i kontinualno će se učitavati, segment po segment. To će sprečiti duže početno vreme učitavanja.



Konfigurisanje muzike za reprodukciju zvuka

Konfigurisanje materijala

Kao finalni korak, sledi konfigurisanje materijala mrežice za modularno okruženje. Prema standardnom podešavanju, ovi materijali su kreirani i konfigurisani automatski u Unityju prilikom importovanja mrežica u panel Project. Obično će biti dodati u poddirektorijum materials, zajedno sa mrežicom. Iz tog poddirektorijuma možemo da prevučemo i otpustimo materijale u direktorijume materijala višeg nivoa u projektu, ujedno organizujući materijale. Ne brinite o pomeranju materijala zbog organizovanja, jer će Unity pratiti sve reference i linkove za objekte.



Konfigurisanje materijala

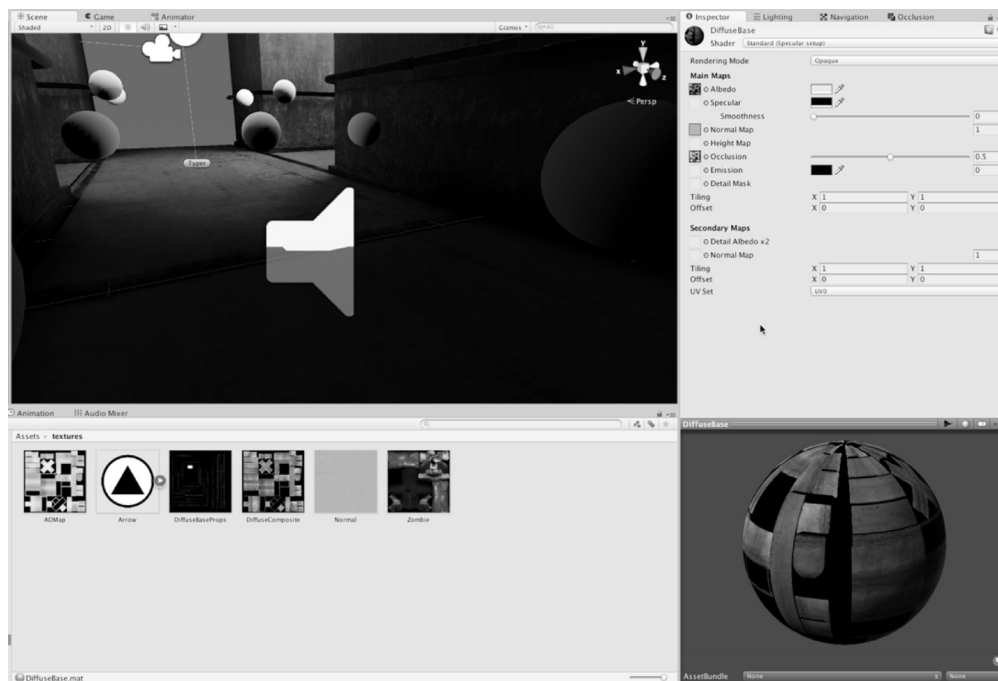
Prema standardnom podešavanju, materijal **DiffuseBase** za modularno okruženje je konfigurisan kao standardni materijal za senke, sa nekim stepenom sjaja. To čini da okruženje izgleda sjajnije i glatkije nego što bi trebalo. Osim toga, materijal nema Normal Map i Ambient Occlusion mapu. Da bismo konfigurisali materijal, selektovaćemo **DiffuseBase** i podesićemo opciju **Shader type na Standard (Specular setup)**:



Menjanje opcije Shader type

POGLAVLJE 1 Priprema i konfigurisanje elemenata postavke

Zatim ćemo dodeliti teksturu **DiffuseBase** texture u slot **Albedo** (glavna difuzna tekstura) i popunićemo polja **Normal Map** i **Ambient Occlusion** dodelom odgovarajućih tekstura koje se mogu pronaći u direktorijumu textures:



Završavanje konfiguracije materijala okruženja

REZIME

U ovom poglavlju razmotrili smo mnogo instrumentalnih koncepata za uspostavljanje solidne osnove za projekat *Dead Keys*. Sada imamo Unity projekat sa importovanom većinom elemenata, koji je spreman za početak rada u sledećem poglavlju. Osnova uključuje do sada importovane module okruženja, mrežice zombija, teksture, audio fajlove i tako dalje. Prilikom importovanja ovih elemenata razmotrili smo mnogo tema, kao što su optimalna konstrukcija elemenata, importovanje smernica i rešavanje uobičajenih i manje očiglednih problema koji se ponekad jave u toku rada. Potpuno pripremljen i konfigurisan projekat, spreman za početak rada, možete da pronađete u fajlovima za ovu knjigu u direktorijumu Chapter02/Start. To će vam poštedeti ručnog importovanja svih elemenata. U sledećem poglavlju fokusiraćemo se detaljnije na projektovanje nivoa i tehnike izgradnje, od skyboxa i osvetljenja, do emocije i atmosfere.

