Izrada 3D igre u Unreal Engine 4 i plasiranje na Steam Marketplace

Rahmani, Sandi

Sveučilište
Sjever

Završni rad br. 470/MM/2016

Izrada 3d igre u Unreal Engine 4 i plasiranje na Steam Marketplace

Sandi Rahmani, 5414/601

Varaždin, lipanj 2017. godine

Sveučilište
Sjever

Multimedija, oblikovanje i primjena

Završni rad br. 470/MM/2016

Izrada 3D igre u Unreal Engine 4 i plasiranje na Steam Marketplace

Student

Sandi Rahmani, 5414/601

Mentor

dr.sc. Andrija Bernik, pred.

Varaždin, lipanj 2017. godine

Sveučilište Sjever
Sveučilišni centar Varaždin
104. brigade 3, HR–42000 Varaždin

UNIVERSITY NORTH

Prijava završnog rada

Definiranje teme završnog rada i povjerenstva

ODJEL: Odjel za multimediju, oblikovanje i primjenu

PRISTUPNIK: Sandi Rahmani | MATIČNI BROJ: 5414/601

DATUM: 19.02.2016 | KOLEGIJ: 3D modeliranje

NASLOV RADA: Izrada 3D igre u Unreal Engine 4 i plasiranje na Steam Marketplace

MENTOR: Andrija Bernik, dipl. inf. | ZVANJE: Predavač

ČLANOVI POVJERENSTVA:

1. mr.sc. Dragan Matković, v. predavač - predsjednik
2. mr.sc. Vladimir Stanisavljević, v. predavač - član
3. pred. Andrija Bernik, dipl. inf. - mentor
4. doc.dr.sc. Winton Afrić - zamjenski član
5.

VŽ KC

MMI

Zadatak završnog rada

BROJ: 470/MM/2016

OPIS

Kroz rad će se objasniti Blender program za 3D modeliranje koji je potreban za izradu modela za 3D igre, zatim kako se koristi Unreal Engine 4 i njegov sustav pod nazivom Blueprints visual scripting, te kako koristeći te programe izraditi jednostavnu 3D igru. Nakon izrade igre u radu će biti objašnjeni koraci kako plasirati igru u web trgovini Steam putem njihovog Greenlight sustava.

Cilj rada je stvoriti dobru referencu za svakog budućeg game developera kako iz prve ruke izgleda posao u toj industriji.

U radu je potrebno:
- opisati Blender i tehnike za 3D modeliranje
- opisati kroz korake proizvodni proces video igre
- opisati Blueprint sustav u Unreal Engine 4
- objasniti odnos između Blendera i Unreal Engine 4
- objasniti postupak plasiranja igre kroz sustav Steam Greenlight
- usporediti Blender sa ostalim programima za 3D modeliranje
- usporediti Unreal Engine 4 sa ostalim game engineima
- usporediti Steam sa ostalim platformama za distribuciju video igara
- prikazati izrađenu igru

REPUBLIKA HRVATSKA
SVEUČILIŠTE SJEVER
KOORDINATOR SVEUČILIŠTE SJEVER

ZADATAK URUČEN: 19.02.2016. | POTPIS MENTORA: Bernik

Predgovor

Danas postoje alati koji pogoduju raznim uvjetima razvijanja videoigre u rasponima kapaciteta studija od jednog do stotina zaposlenika. S ciljem da, zaljubljenicima i onima koji vide svoje buduće radno mjesto u ovoj branši, približim proces izrade videoigara uz pomoć dva programa koji su u vrijeme pisanja ovog rada širokonamjenski i s kojima bi se netko tko se nađe u ovoj industriji najvjerojatnije susreo, Blender i Unreal Engine 4.

Sažetak

U radu su predstavljeni alati kojima se netko može služiti za izradu videoigara. Besplatni alat otvorenog koda za 3D modeliranje Blender i odnedavno gotovo besplatan program za izradu videoigara, takozvani game engine Unreal Engine 4 te su oba uspoređena sa konkurentim programima u njihovim područjima. Također je predstavljena i platforma Steam i objašnjen sustav plasiranja igara na tu platformu u kontektsu ostalih platforma za plasiranje igara. Za bolje razumjevanje Unreal Engine 4 i za svrhu stvaranja reference za svakoga tko se želi okušati u izradi videoigara prikazan je primjer izrade jednsotavne 3D videoigre.

Ključne riječi: Blender, Unreal Engine 4, Steam, Videoigre

Summary

This project presents some tools someone can use to make a videogame. A free open source tool for 3D modelling Blender and the game engine Unreal Engine 4 which is free since recently. Both are compared to their respective competitors. Furthermore the platform Steam is presented and the process of publishing to the platform is presented in the context of videogame publishing platforms. For a better understanding of Unreal Engine 4 and for the purpose of creating a reference for anyone who'd like to try making a videogame an example of creating a simple 3D videogame is also presented.

Keywords: Blender, Unreal Engine 4, Steam, Videogames

Popis korištenih kratica

UE4	Unreal Engine 4
VR	Virtual Reality
PSN	Play Station Network
UX	User Experience
FPS	Frames Per Second
AI	Artificial Intelligence

Sadržaj

1. Uvod 1
2. Blender 2
3. Unreal Engine 4 5
   3.1. Blueprints 6
4. Steam 7
   4.1. Konkurencija 8
5. Od Blendera do UE 4 9
   5.1. Mjerne jedinice 10
   5.2. Orijentacija 10
   5.3. Teksture i materijali 10
   5.4. FBX export 10
   5.5. UE4 import 11
6. Od UE4 do Steama 12
   6.1. Steam Greenlight 13
   6.2. Steam Direct 14
7. Proces razvoja videoigre 15
8. Primjer razvoja igre 17
9. Zaključak 38
10. Literatura 39
11. Popis slika 41

1. Uvod

U današnje vrijeme industrija videoigara je golema. Ukupni prihodi od prodaje videoigara su 2016. bili veći od $90 milijardi. Ta činjenica sama govori koliko su veliki aspekt svakodnevnog života postale videoigre.[12] Iako je industrija danas ogromna nije oduvjek bilo tako, desetljeće ili 2 unazad igre su bile rezervirane za specifične grupe ljudi no pojavom interneta i uspješnih platforma za distribuciju poput Steama te pojavom smartphonea i mobilnih igara sve više ljudi ima pristup i interes za ovu specifičnu vrstu interaktivne razonode.

Videoigre su pridonjele i velikom napretku u računalnoj grafici kako kroz razvoj novih, bržih i ljepših prizora kako bi bili korak ispred konkurencije tako i kroz razvoj takozvanih game engina. Game engine je širokopojasna platforma za spajanje svih elemenata koji čine jednu videoigru u jednu cjelinu. Danas ih ima mnogo, a jedan koji zadnjih godina dobiva na popularnosti, djelom zbog toga jer je postao besplatan a djelom i zbog toga što koristi nešto što se zove „visual scripting“ kroz svoj Blueprint sustav je Unreal Engine 4. Visual scripting je način programiranja gdje programer ne piše kod nego spaja predefinirane elemente koda veoma nalik na shemu. Sve više engina prelazi na takav vizualni tip rada zbog toga što to otvara vrata mnogim developerima koji nemaju iskustva u programiranju ili ih odbija takav način rada.[20]

U posljednih nekoliko godina pojavilo se podosta dobrih game engina koji se orijentiraju na male nezavisne studije što je djelom pridonjelo tome da igre postanu tako proširene i popularne jer je to značilo da game developement više nije rezerviran za velike kompanije s divovskim buđetom i radnom silom od više desataka ili stotinja ljudi.

2. Blender

Blender je besplatan 3D program otvorenog koda.[1] On je prvi i jedini u potpunosti integriran paket za stvaranje 3D grafike koji se može koristiti za modeliranje, animaciju, renderiranje, post-produkciju, stvaranje i pokretanje interaktivnih igara i 3D-a u stvarnom vremenu te posjeduje višeplatformsku kompatibilnost.

Napredni korisnici mogu koristiti Python da prilagode Blender vlastitim potrebama pišući specijalizirane funkcije za izvođenje specifičnih zadataka. Često se takve funkcije pronađu u budućim verzijama Blendera. Odlično odgovara freelancerima i malim studijima zbog unificiranog workflowa i dinamičnosti same aplikacije.

Sučelje koristi OpenGL, a projekt je vođen pod GNU General Public Licencom što znači da svatko ima pravo raditi male ili velike promjene na kodu što dovodi do novih mogućnosti, brzih popravaka za greške i boljeg korisničkog iskustva.[28] Blender je besplatan ali svatko je pozvan da investira ili sudjeluje u razvijanju ovog besplatnog alata.

U usporedbi s drugim programima slične namjene Blenderova najveća prednost je to što je besplatan. Budući da je program otvorenog koda to podrazumijeva da veliki broj ljudi volonterski radi na njegovom usavršavanju. S jedne strane rezultat toga su česta izdavanja novih verzija s novim mogućnostima i alatima u koraku s aktualnim potrebama, dok s druge neki ljudi instinktivno izbjegavaju programe otvorenog koda jer su drugi iste kategorije nedorasli svojim komercijalnim konkurentima pa se oko te titule stvorio osjećaj skepticizma. Najviše su mu privučeni ljudi koji tek započinju s 3D-om i oni koji to žele raditi iz zabave i hobija. Danas se rjeđe koristi u velikim studijima koji preferiraju zaposlenike s višegodišnjim iskustvom pa takvi najčešće imaju već dobro utabano iskustvo u programima kao što su Maya i 3ds Max. Kontrastno tome mali i novi studiji puno češće koriste Blender zbog toga što je besplatan što uvjek pomaže kada se kreće u nove poslovne pothvate.

Prilikom pokretanja programa pojavljuje se plutajući prozor. Uloga plutajućeg prozora je omogućavanje brzog i jednostavnog pristupa prethodno otvorenih Blender datoteka, te pristup korisnim linkovima. Ako se želi krenuti iz početka s novom datotekom potrebno je kliknuti mišem izvan tog prozora. U tom slučaju on nestaje i korisniku se prikazuje osnovna scena koja sadržava kocku, lampu i kameru te je sve spremno za početak rada.

Blender je napravljen na takav način da se nalazi samo u jednom modu istovremeno. Svaki od modova dizajniran je posebno za uređivanje određenih aspekata objekta. Dva osnovna moda ili načina rada su editorski (eng. edit) mod te objektni (eng. object) mod.[2] Pri pokretanju programa korisnik se po osnovnim podešenjima nalazi u object mode-u. Prebacivanje iz jednog moda u drugi radi se klikom miša na dnu 3D view prozora, na padajući izbornik koji tada nudi nekoliko različitih modova uključujući i edit mode. Kao što i većina funkcija ima svoj prečac (eng. hotkey) tako ima i ova funkcija te se pritiskom tab tipke na tipkovnici može puno brže i jednostavnije prebaciti iz jednog moda u drugi.

Objekti se mogu odabirati samo u object mode-u i on služi za manipulaciju objekata u cijelosti, npr. rotiranje, skaliranje, te pomicanje kompletnih poligonalnih mreža, dok se u edit mode-u vrši manipulacija pojedinih dijelova objekata. Edit mode se odnosi isključivo na samo jedan objekt i to samo na aktivni objekt, ili na zadnje odabrani. Manipulacija se može vršiti na točkama poligona (vertice, tj. eng. vertex), rubovima poligona (eng. edge) te na samim poligonima (eng. face) omeđenim sa tri ili više točaka

Ovisno o potrebama rada Blender nudi mogućnost odabira različitih, unaprijed smišljenih postavki. Naime, korisniku je omogućeno biranje postavki za modeliranje, animaciju, kompoziciju, skulpturiranje, video uređivanje, UV editiranje, te kreiranje videoigara. Ako korisnik ne želi koristiti unaprijed smišljene postavke za određenu vrstu rada, može sam prilagoditi radnu okolinu. Svaki radni prozor nudi opciju dijeljenja pa se tako jedan radni prozor može podijeliti na nekoliko njih. To se postiže klikom lijeve tipke miša u gornjem desnom uglu prozora i povlačenjem u stranu, suprotno od ruba.

Prema osnovnim postavkama u object mode-u odabrani objekt se prikazuje narančastom bojom te također u edit mode-u. Postoji mogućnost promijene boje pritiskom na File, zatim User Preferences te na Themes. U object mode-u prilikom žičanog prikaza modela odabrani model će također biti prikazan narančastom bojom, dok će ostali biti prikazani crnom bojom. Ukoliko se odabere nekoliko objekata svi selektirani objekti, osim aktivnog koji je zadnji odabran, biti će prikazani tamno narančastom bojom. Ista je stvar i u edit mode-u. Odabrane točke, rubovi ili poligoni biti će narančaste boje. Treba samo napomenuti da je aktivni poligon označen bijelom

bojom. Kao što je već ranije spomenuto, u edit mode-u se samo jedna poligonalna mreža može editirati odjednom. Međutim postoji mogućnost spajanja više objekata u jednu poligonalnu mrežu. To se ostvaruje selektiranjem objekata, te pritiskom tipki CTRL + J u object mode-u, a ukoliko se žele razdvojiti objekti tako da svaki ponovno čini zasebnu poligonalnu mrežu potrebno je odabrati željenu poligonalnu mrežu u edit mode-u te pritisnuti tipku P. Ukoliko su odabrane dvije susjedne toče u edit mode –u postat će narančaste boje i rub koji spaja te dvije toče također će postati narančaste boje. Isto tako cijeli poligon će također postati narančast ukoliko se odabere dovoljan broj točaka ili rubova da zajedno tvore poligon.

Za dodavanje novih objekata potrebno je kliknuti na Add u izborniku u zaglavlju ili pritisnuti kombinaciju tipki SHIFT + A unutar 3D view prozora. Tada se nude razne opcije, na primjer geometrijski primitivi (kocka, kugla i sl.), krivulje, tekst, armatura, kamera, rasvjetna tijela te mnoge druge opcije. Ukoliko se želi obrisati neki objekt iz scene potrebno ga je selektirati desnim klikom miša u object mode-u te pritisnuti tipku DELETE ili X. Prema osnovnim postavkama Bleder koristi solid tehniku sjenčanja. U ovom modu modeli su prikazani sivom bojom bez tekstura. Osim ovog moda moguće se prebaciti u takozvani žičani pogled (eng. wireframe) i u tekstured pogled. Tekstured pogled prikazuje teksture objekata i omogućuje korisniku da pomicanjem rasvjete po sceni vidi utjecaj osvjetljenja na model odnosno svjetlosne promjene na modelu u stvarnom vremenu. Prebacivanje je moguće napraviti u izborniku na dnu 3D view prozora, a za brzo prebacivanje između solid i wireframe pogleda koristi se tipka Z.

Za transformiranje objekata u sceni, Blender nudi dodatak pod imenom Transform Orientation. To je vrlo koristan dodatak jer se u njemu određuje u kojem se koordinatnom sustavu korisnik nalazi odnosno koje će se koordinatne osi koristiti. Svaka koordinatna os je prikazana drugačijom bojom. X os je crvena, Y os je zelena, a Z os je plava. Prema osnovnim podešenjima, Blender se prilikom pokretanja nalazi u globalnom koordinatnom sustavu. Kad je odabran jedan sustav, on se ne mijenja bez obzira na promjenu objekata u sceni ili promjenu pogleda. Mijenja se tek kada korisnik iz izbornika Transform Orientation odabere drugačiji sustav. Osim globalnog sustava moguće je selektirati lokalni sustav koji pokazuje orijentaciju objekta koji je selektiran. Također jedan od korisnijih sustava je view koordinatni sustav u kojem su osi X i Y uvijek okomite na smjer iz kojeg korisnik gleda 3D prostor. Na taj način bez obzira na pomak kamere odnosno pogleda korisnik ima mogućnost pomicanja objekata okomito na smjer gledanja. Osim promjene pozicije u prostoru objekti se mogu i skalirati ili rotirati po koordinatnim osima.

3. Unreal Engine 4

UE4 je program koji je razvijen od strane Epic Games studija koji su tvorci videoigri Unreal Tournament, i serijala igri Gears of War i Infinity Blade. [6] Trenutno rade na nekoliko naslova pod imenima Paragon, Fortnite, SPYJiNX, Battlebreakers, RoboRecall i novu instalaciju Unreal Tournamenta. Danas je jedan od najjaćih game enginea i može se korsititi za izradu igri za mobitele, računala svih operacijskih sutava, konzole i VR.

Prije je program bio rezerviran samo za velike studije jer je licenca za korištenje dosezala više tisuća dolara.[14] Relativno nedavno Epic Games je odlučio promjeniti strategiju i sada nude UE4 besplatno na korištenje svima koji žele s njim učiti i raditi dok ne plasiraju svoju igru na tržište kada poćinju plačati 5% prihoda od prodaje igre .[15] Konkurira Unity Engineu koji je i dalje najzastupljeniji game engine na tržištu među nezavisnim studijima zbog svojih brojnih tutoriala na internetu, velike zajednice koja si međusobno pomaže, lakoće korištenja i besplatnih komponenti za izradu igre koje se mogu naći na internetu.[16] Unity je do sada bio praktički jedino rješenje za nekoga tko se htio početi baviti razvojem 3D igara i s jedne strane je stvoren loš dojam o Unityu kao programu koji stoji iza loših igara jer su se sve i one dobre i loše igre radile u njemu. Pridodati tome situaciju s licenciranjem gdje je netko na najslabijoj licenci morao postaviti Unity logo na početni zaslon kod pokretanja igre dok one jače licence to ne zahtjevaju. S pretpostavkom da onaj koji ima slabije resurse za licence ima i slabije resurse za izradu igre na internetu se pojavilo mnogo igara upitne kvalitete koje moraju staviti Unity logo na početni zaslon i neke jako kvalitetne igre koje nisu stavljale Unity logo, lako je vidjeti zašto je postao omražen kod nekih igrača. UE4 nema taj problem jer su sve igre do sada izrađene u njemu visoke kvalitete budući da opseg igara izrađenih u njemu većinski dolazi od visokobuđetnih studija dok trenutno oni nezavni tek uče koristiti program jer nitko od njih nije imao pristup da bi usavršio vještine korištenja. Epic Games ulaže mnogo truda kako bi educirao nove korisnike sa velikom detaljnom dokumentacijom svih djelova programa, dugom listom detaljnih i opsežnih video tutoriala na Youtubeu, učestalih, višesatnih, edukativnih video streamova i veliku bazu demonstrativnih projekata videoigri koje svatko može koristiti i za komercijalne projekre bez obaveza.

Između ekskluzivne visokobuđetne licence i gotovo besplatnog statusa postojao je tranzicijski period kada su nudili pretplatnički model od 20 dolara mjesečno.[17] Nakon što je 2. ožujka 2015. postao besplatan izdali su povrat novca za prošli mjesec pretplate i poklon od $30 kredita na trošenje u Marketplaceu. Marketplace je Epicov odgovor na Unity Asset Store gdje developeri mogu kupovati ili postavljati komponente videoigri i neprestano raste.

3.1. Blueprints

Starije verzije Unreal Enginea radile na bazi njihovog UnrealScript koda. Kasnije je uveden sustav sličan današnjem Blueprintu pod nazivom Kismet no nije bio toliko ispoliran i napredan kao što je to Blueprint danas.

Blueprint je sustav vizualnog skriptiranja unutar UE4. Vizualno skriptiranje podrazumjeva način stvaranja logike pomoću grafičkih reprezentacija tekstualnog koda.[7] U ovom slučaju vizualno skriptiranje je izvedeno pomoću blokova koji se spajaju nalik na shemu. Odnosi između komponenata regulirani su krivuljama koje se spajaju na sidrišta na grafičkim elementima komponenti. Razne vrte komponenti kategorizirane su po raznim bojama pa čak postoji i sustav koji se nadovezuje na prethodnu komponentu i ponudi sve moguće komponente koje odgovaraju i mogu biti spojene na to sidrište da nebi dolazilo do greški u kodu. Sustav Blueprintsa je daleko prevelik da bi ga se obradilo u ovom radu ali Epic Games nudi potpunu dokumentaciju o svakom elementu Blueprints sustava na svojim stranicama.

Od kad je UE4 napustio UnrealScript ponudila se opcija kodiranja i u C++ i kasnije je implementiran konvertor koji je mogao C++ kod prevesti u Blueprint a nedugo prije pisanja ovog rada pojavila se i mogućnost pretvaranja Blueprinta u C++ kod.[18] Ovo je bio veliki pomak jer to omogućujugotovo istanjenje granice između onih koji više naginju na vizualne aspekte razvoja igre i onih koji su čisti programeri. Sada umjetnici mogu testirati prototipe sa Blueprintima i ne zamarati programere sa sitnicama dok se programeri mogu fokusirati da kod radi kako treba, a dvosjmerna koverzija logike iz C++ i Blueprinta omogućava uzimanje onih stvari koje funkcioniraju iz oba svjeta i brže koračanje prema završavanju projekta.

Doduše treba uzeti u obzir činjenicu da je Blueprint sustav jedan sloj iznad koda tako da nemože nikad biti toliko čist i fleksibilan kao čisto pisanje koda i u današnjom stanju procjena je da je Blueprints sustav oko 10 puta teži na računalu od čistog koda za istu logiku što je izuzetno dobro za takve sustave i uzevši u obzir da je nedavno ta brojka bila oko 100.[18]

4. Steam

Valve je sjevernoamerička tvrtka koja se bavi izradom softwarea. Valve se proslavio svojom prvom igrom Half-Life koja je bila revolucionarna za to vrijeme te se i dan danas bave izradom videoigara ali primarno održavanjem Steam platforme.

Gabe Newell je osnovao Valve 1996.[13] kada je napustio microsoft i kao već tadašnji multimiljunaš osnovao Valve. Kompanija svoj uspjeh možda duguje prvenstveno filozofijom kojom je Newell išao u posao, a to je uzimanje ljudi sa samog vrha specifičnog područja bez obzira na cjenu sa pretpostavkom da je njihova visoka cjena opravdana.

Valve je također jedna od kompanija za videoigre koja slovi kao jedna od najboljih u kojoj se može pronaći posao zbog svojih radnih uvjeta.[4] U Valveu ne postoje direktori i voditelji odsjeka već je hijerarhija među radnicima horizontalna što znači da je svaki radnik ravnopravan sljedećem i nitko mu nemože odrediti što točno mora raditi. Svaki radnik dobi svoju radnu stanicu uključujući i stol sa kotačima koji radnici pomiću po uredu, spajaju se u grupe i rade na projektima za koje se dogovore. Primjerice jedan radnik dobije ideju i kreće u komunikaciju sa kolegama u nastojanju da ih uvjeri da mu pomognu raditi na toj ideji. Iako se možda takav sustav na prvi pogled čini kaotičan oni su očuti pronašli način da to funkcionira. Također nude priručnik za sve nove zaposlenike ili one koji to tek planiraju postati.

Svoju prvu igru, Half-Life, po kojoj su i danas poznati izradili su na Source game engineu koji razvili na bazi Quake enginea. Kasnije igre kao što su Half-Life 2 , Portal i Left 4 Dead su radili na novoj verziji enginea koji su nazvali Source 2 koji je bio puno popularniji i ljudi su ga u govoru nazivali jednostavno Source pa se danas originalni Source engine naziva Source Gold ili Source Classic kako bi se izbjegla zabuna između njih.[13]

Nedugo nakon Half-Lifea zaljubljenici u igru pronašli su način kako napraviti modove za igru. Iz toga je proizašao prvi mod za Half-Life pod nazivom Counter-Strike na kojeme je radilo nekoliko fanova iz razonode. Nakon što je Valve uvidio potencijal za igrom otkupili su prava na naslov i cijeli tim developera koji je u to vrijeme radio na njemu te su izbacili samostojeću igru koja je nakon nekoliko iteracija posljednji update dobila pod brijem verzije 1.6 pa se danas ta vezija originalnog Counter-Strikea naziva Counter-Strike 1.6.[19] Tim trendom je Valve nastavio raditi i sljedećih godina kada bi vidjeli tim ljudi koji radi na nekim mehanikama za igru koja im se sviđa otkupili bi cjeli tim da radi za njih i plasirali igru. To je bio slučaj sa Left 4 Dead, Dota i Portal franšizom. Team Fortress i Half life su razvijani interno od početka.

Steam je nastao iz potrebe za jednostavnijim i bržim načinom slanja novih verzija Counter-Strikea igračima. Polako se sustav širio i dodavano je sve više igara koje su imale benefit od platforme koja može dopremiti update instantno preko interneta i svima u isto vrijeme. To je dovelo do svojevrsne standardizacije svake individualne igre jer se proizvođač nemora baviti s višestrukim verzijama igara koje postoje i uzimati u obzir njihove razlike kako bi pružio čim bolju podršku igračima nego se samo bavi aktualnom verzijom igre i sve je na jednom mjesu. U početku su to bile samo Valve igre no kasnije su otvorili vrata i drugim studijima te pretvorili Steam iz platforme za distribuciju updatea u platformu za distribuciju igara i društvenu mrežu. Danas Steam nudi više od 1800 igara u svojoj trgovini i ima više od 35 milijuna aktivnih korisnika.[5] Steam je danas postao glavni fokus cjelog Valvea i već 5 godina od izlaska DOTA 2 nisu izbacili nijednu samostojeću, originalnu igru. 2017. je najavljena igra Artifact koja je predstavljena kao DOTA igra kartama na opće neoduševljenje publike.[21] Valve se u posljednje vrijeme fokusira na VR igre što je vidljivo kroz cjeli spektar njihovih usluga.

4.1. Konkurencija

Valve za korištenje njihove digitalne distribucijske platforme Steam naplaćuje 30% od ukupne prodaje.[22] Nekim studijima i izdavačkim kućama se to nije svidjelo pa su išli u razvijanje svojih vlastitih platforma. Najveći od njih danas su EA koji ima svoju Origin platformu, Ubisoft sa Uplay platformom i Microsoft sa Windows Storeom. Oba na svojim platformama nude svoje vlastite proizvode i neke druge od studija s kojima su sklopili ugovor kao i Steam no najčešće u puno manjim opsezima korisnika i ponude. Neki od njih drže svoje proizvode ekskluzivnima za te platforme, odnosno ne žele ih staviti na Steam trgovinu zbog različitih razloga. U međuvremenu su se pojavile i platforme koje žele direktno konkurirati Steamu no nisu bile uspješne. Jedna od značajnih je bila Desura koja je imala svoj klijent i web stranicu i nudila igre po istim ili nižim cjenama od Steama s manjim postotkom za distribuciju za developere no nisu uspjeli opstati. Danas se najčešće spominju Itch.io i Humble Store. Oba su izgradila svoje platforme za istu ciljanu publiku ali na različite načine. Itch.io je primarno web stranica koja ne naplaćuje ništa i prima sve i svakoga u svoj sustav tako da mali developeri vole koristiti platformu za testiranje igri i interesa ljudi za igru jer nema uključenog rizika u tome. Humble Store je relativno novi ogranak Humble Bundlea koji je dobrotvorni Bundle servis. Bundle znači da se više od jedne igre zapakira u jednu cjelinu po nižoj ukupnoj cjeni od ukupne individualne cjene svake igre u paketu. Humble Bundle je najpopularniji od takvih i sa Storeom nudi istu digitalnu trgovinu kao i drugi.[23]

5. Od Blendera do UE 4

U procesu izrade videoigara korisnik treba biti spretan u svim aspektima Blendera s iznimkom ugrađenog game enginea koji se jako rjetko koristi u praksi zbog upitne kvalitete. Kod izrade igre tjek rada (eng. workflow) se razlikuje po potrebi igre i studija no u praksi se često vidi određena podjela po jačini studija i načinu rada no nijedna nije isključiva i oba koriste obje tehnike. Jedan način, koji je češće viđen kod jačih studija, započinje sa sculptingom. Prvo se radi detaljna skulptura modela kao od gline. Nakon što je model spreman počinje ga se bojati teksturom i materijalima. Taj dio je moguć direktno u Blenderu ali većina preferira koristiti eksterne programe koji su specifično namjenjeni za to. Nakon ili prije ovog procesa za model se mora izraditi kostur. Važno je napomenuti da se kod modeliranja za videoigre modeli antropomorfnih likova izrađuju u takozvanim A i T pozama. T poza je poza lika gdje su mu ruke ispružene vodoravno okomito na noge koje su paralelne dok A poza stavlja ruke lika pod kut manji od 90 stupnjeva u odnosu na tjelo. Prsti i ostali ekstremiteti su potpuno ispruženi na modelu. Ovo je važno jer to jako olakšava povezivanje kostiju s željenim djelom 3D modela.

U procesu izrade videoigara model se ne renderira jer se parametri kod materijala ne prikazuju jednako u game engineu a postavke scene, osvjetljenja i ostali parametri koji nisu dio modela se uopće ne prenose. Razlog tome je što game engine ima svoj vlastiti render engine koji ima svoje postavke i drugačije prikazuje grafiku. Iznimka ovome je što se neki jednostavniji materijali s najosnovnijim postavkama slično prikazuju u Blenderu i u Unreal Engineu pa je moguće to koristiti kroz import export.

Drugi workflow koji se češće vidi u manjim studijima počnje sa modeliranjem na klasični način bez skulpturiranja. Razlog tome je što manji studio ima manje resurse pa jedan čovjek mora odraditi veći opseg posla nego što je to u većem studiju. Rezultat toga je što se manji studiji češće idu putem low poly grafike jer na taj način mogu uštedjeti vrijeme. Manje poligona na modelu znači manje vremena na detaljima, manje kostiju i manje detaljne teksture. Nakon što se izradi model i postavi kostur može se ići putem bojanja teksture i materijala na model ali kod low poly model češći slučaj je da ti modeli imaju i niže rezolucije pa se to nekako uklopi u stiliziranu grafiku cjele igre. U tom slučaju radi se UV mapa i eksporta se u neki grafički program kao što je Photoshop gdje se s njegovim alatima nacrta tekstura te se natrag importa u Blender da se podesi UV mapa po potrebi da svaki piksel sjedne na svoje mjesto. U nekim slučajevima, kako bi još više ušparali na vremenu, modeli imaju samo jednobojne materijale bez tekstura na sebi što može također dati dobar stilizirani izgled igri.

5.1. Mjerne jedinice

Osnovna stvar koju treba prilagoditi za lakši worflow su mjerne jedinice.[3] Jedna jedinica mjere u UE4 se prenosi u 1cm. Da se Blender prilagodi toj skali mora se u Properties panelu pod Scene izbornikom postaviti Units na Metric i Scale na 0.010. Sada će se sve dimenzije modela u Blenderu prikazivati u centimetrima i jedan kvadrat podne matrice u Blenderu je jednak jednom kvadratu podne matrice u UE4.

5.2. Orijentacija

Sljedeća stvar na koju treba obratiti pažnju je orijentacija. Blender kao naprijed smatra negativnu stranu Y osi dok UE4 pod naprijed smatra pozitivnu stranu X osi. Tomu se može prilagoditi na više načina. Jedan je da se model od početka radi tako da mu je prednja strana okrenuta prema pozitivnom djelu X osi ali to može predstavljati male zapreke ako se koriste predefinirane rotacije i pozicije kamere. Drugi način je da se po završetku rada model okrene za 90 stupnjeva po Z osi no to isto može predstavljati probleme ako se neki parametri postavljaju relativno na osi pa može doći do distorzije modela. Posljednji način koji je možda i najpreporučljiviji je da se kod exporta modela odabere koja os predstavlja naprijed. Tako originalna verzija modela ostane nepromjenjena nego samo novo stvorena datoteka ima prilagođenu rotaciju za UE4.

5.3. Teksture i materijali

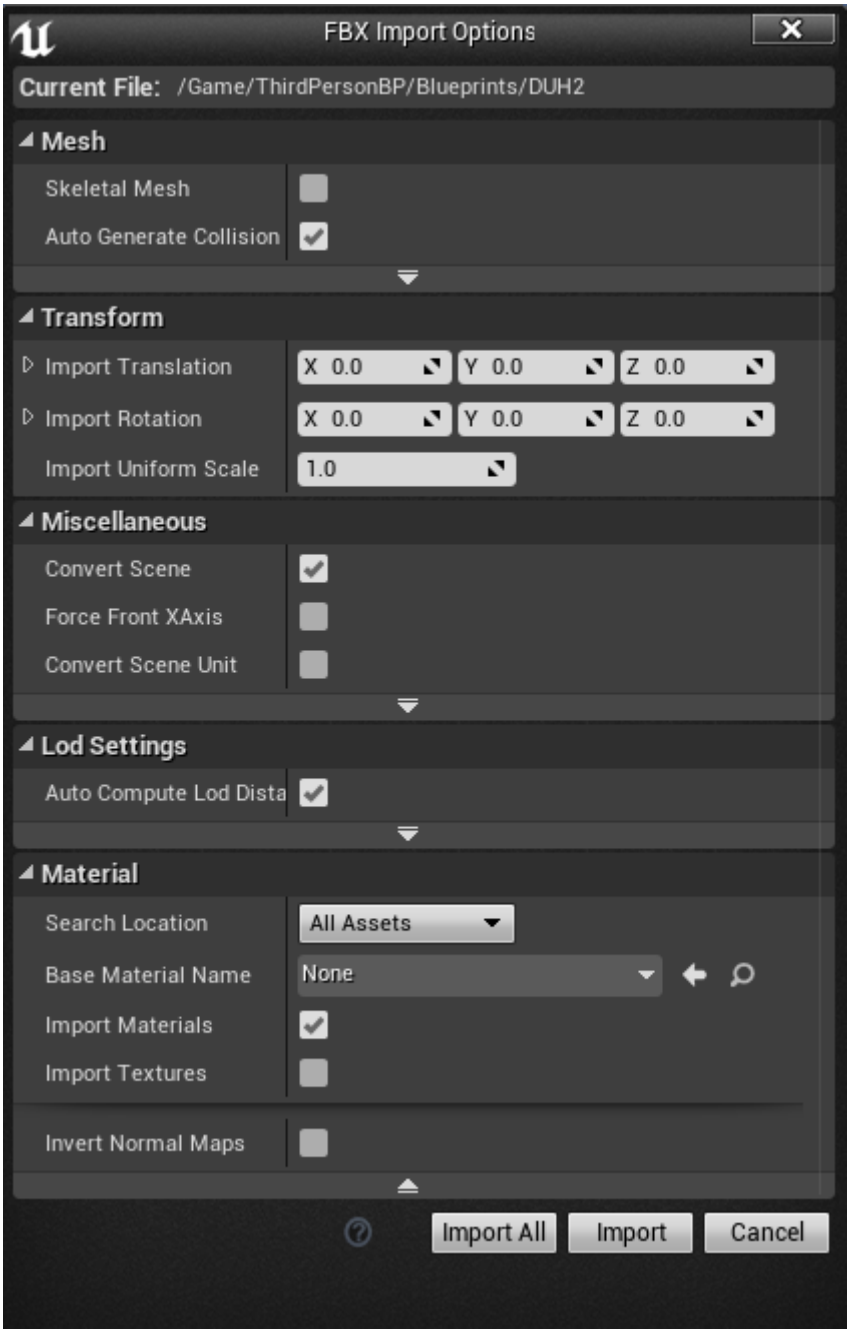
Prije nego se napravi tekstura ili aplicira materijal mora se napraviti UV mapa kako bi se isti mogli prenjeti iz Blendera u UE4. UV mapa u Blenderu će dati UE4 informacije o tome kamo smjestiti svaki piksel teksture. U slučaju da se ne koriste teksture za modele u UE4 mogu se postaviti materijali po željenim područjima na modelu i UE4 će prepoznati koja su to područja na modelu pa se na ista područja može primjeniti neki materijal iz UE4, no bez teksture. Ako se pokuša primjeniti teksturu na model bez UV mape to područje neće biti obojano teksutrom nego jednobojno.

5.4. FBX export

Modeli se od Blendera do UE4 prenose u .fbx formatu što je najčešće slučaj i kod drugih 3D i game engine kombinacija. Kod exporta postoji nekoliko opcija koje su specifična za razne potrebe.[24]

5.5. UE4 import

Unreal Engine ima dobar sustav importanja. Prilikom toga odabere se generalno import pa program sam prepozna prema datoteki koja se odabere kakvo će to prevesti u engine. Kada se odabere željena datoteka, u slučaju modela iz Blendera .fbx, korisniku se nude razne opcije na izbor. Primjerice korisnik može uvesti samo neke modele, kombinirati ih sve u jedan ili ih zadržati razdvojene ako su razdvojeni u originalnoj datoteci u Blenderu, može se koristiti samo kostur, generirati kolizija za model, birati hoće li se koristiti materijali iz Blendera ili ne, veličina modela i još druge stvari.[24]



Slika 5.1 UE4 FBX Import Options

6. Od UE4 do Steama

Prije nego se krene sa samom izradom videoigre važno je odlučiti na koje platforme će proizvod biti plasiran. Oni koji se tek počinju baviti s izradom videoigara najčešće biraju plasirati na računalo zbog toga što je platforma, računalo, na kojoj će se igra igrati ista platforma na kojoj se igra i razvija. Pri tome se štedi na vremenu i poslu jer se nemora imati na umu koje mogućnosti možda neće dobro raditi na drugim platformama i nemora se prilagođavati sadržaj igre u ravoju na platformu za testiranje iako je danas u nekim programima to jako kvalitetno rješeno pa je testiranje na drugoj platformi, naprimjer mobilnim uređajima, dobro integrirano u razvojni proces. Još jedan razlog zašto mnogi biraju razvijati igre za računala je činjenica da je Steam najjača računalna platforma i plasiranje igre na Steam je jako prijateljski orijentirano prema svim kalibrima studija za razvoj videoigara. Kada se pogledaju procesi kroz koje netko mora proći da plasira svoju igru na druge platforme kao što su to PSN ili Apple Store vidljivo je zašto je Steam danas najpopularnija opcija.

Mobilne platforme kao što su Apple Store i Android imaju naknade koje se plačaju po igri koja se postavlja na njihovu platformu. Apple je skuplji sa cjenom od $99 po aplikaciji no nudi više mogućnosti dok je Google Play jeftiniji sa $25. Razlika je što Apple Store ima interni tim koji ručno odlučuje koje će aplikacije ići u Store i promovira aplikacije na dnevnoj bazi za koje misle da zaslužuju biti više viđene. Google Play je s druge strane imao potpuno slobodan sustav do 2016. Nije nikakve kontrole prije nego aplikacije dolaze na Play osim ako se kasnije utvrdi da krše uvjete bile su maknute. Od 2015. i na Play postoji period čekanja gdje se provjerava dali aplikacija krši pravila.[26]

Platforme za igraće konzole se razlikuju no sve su slične po nekim karakteristikama. Najveća razlika kod razvijanja igre za konzole u usporedbi sa razvjanjem za računalo je ta što onaj koji razvija igru treba imati takozvani Dev Kit. Dev Kit je specijalna verzija konzole koja ima jače komponente i prilagođeni software za testiranje i razvoj igara. Dev Kit se nemože kupiti iz prve ruke nego se mora zatražiti suradnja od tvrtke za čiji sustav se želi razvijati i mora im se prezentirati projekt na kojem studio želi raditi te po odobrenju oni šalju Dev Kit.[25] Velika zamjerka platformama za konzole na koju se i sam Gabe Newell požalio kada su izdavali svoje igre na konzole je sporost kod ažuriranja novih inačica igre. To je još jedan obruč kroz koji se mora preskakati kod razvoja za konzole. Naprimjer u splučaju sa Play Stationom svaka nova verzija igre koliko god mala bila mora biti testirana od strane Sonya pa se tek onda pušta do igrača. Taj proces može trajati i po nekoliko mjeseci pa je ažuriranje igre dosta ograničano.

6.1. Steam Greenlight

Najjača platforma za računalne igre, Steam, ima jako dobro rješen sustav za plasiranje. Od 30. kolovoza 2012. taj sustav se zvao Steam Greenlight i on je ugašen 6. svibnja 2017.[9]

Ovaj sustav je radio na temelju glasanja. Svi krosinici Steama su mogli pristupiti sustavu Greenlight i glasati za igre koje žele vidjeti u Steam Store trgovini. Tko god je želio plasirati igru na Steam morao je platiti jednokratnu naknadu od € 100 nakon koje se moglo plasirati neograničen broj igara na glasanje sa tog računa. Nakon plaćanja naknade račun sa kojim je vezana uplata dobiva puni pristup Steamworks sustavu. Steamworks je sučelje kojem se pristupa kroz internet preglednik sa svojim jedinstvenim korisničkim imenom i lozinkom. Kroz to sučelje korisnik ima mogućnosti upravljati svim značajkama svoje igre kao što su popusti, generacija aktivacijskih kodova, integracija Steamovih servisa u igre kao što su Achievements i Trading Cards, uvid u prodaju, postavljanje grafičkih elemenata na stranicu u trgovini i druge stvari.

U bilo kojoj fazi razvoja igre mogla se postaviti stranica za glasanje na Greenlight. Prije postavljanja trebalo je imati na umu koliko sadržaja se trenutno moglo pokazati korisnicima koji će glasati kako bi glasali pozivitno. U slučaju da se stranica postavila prerano u procesu proizvodnje i predstavljene slike i videi se nisu dojmili korisnicima glasali bi negativno te se igri smanjila vjerojatnost dolaska u trgovinu. S vremenom kako se postavljalo i nekoliko desetaka igri dnevno na Greenlight prva 2 dana su bila ključna za sakupljanje glasova jer se kasnije igra izgubila u moru drugih.

Igra bi došla u trgovinu tek kad bi je interni tim u Valveu odobrio. To je značilo da uz sustav glasovanja igra mora pridržavati i nekih uvjeta tako da neke igre nebi izazvale kontroverzu i dovele Valve u nepoželjan položaj. U slučaju igara koje je interni tim u Valveu smatrao izuzetno dobrima, one su dobivale zeleno svjetlo vrlo brzo s nekolicinom glasova. S druge strane takva dinamika je stvorila i slučajeve gdje je igra imala veliki broj pozitivnih glasova no ni nakon godine dana čekanja nije prošla u trgovinu. Najčešći slučaj kod igara koje su prošle kroz Greenlight je bio da sakupe dovoljan broj glasova i od tih glasova dovoljan broj pozitivnih da bi se našle u top 10 igara na Greenlightu u nekom trenutku.

U sustavu Greenlight bilo je dosta rupa koje su neki iskorištavali pa je bilo slučajeva kupovanja glasova gdje bi se na nekoj drugoj stranici postavila nagradna igra za čije sudjelovanje je potrebno glasati pozitivno za neku specifičnu igru za koju tvorac nagradne igre ima interesa, najčešće postotak od zarade. Valve bi nakon otkrivanja slučajeva gdje se to događalo izbacio igre s trgovine i zabranio daljnje poslovne odnose sa počiniteljima.

6.2. Steam Direct

Steam Direct je novi sustav za plasiranje igara na Steam koji je postao aktivan 13. svibnja 2017., odnosno tjedan dana nakon gašenja Greenlighta. Ovim sustavom se pokušava zaobići taj problem s lažnim glasanjem jer je sustav glasanja potpuno izbačen.[10]

Steam Direct funkcionira na način gdje se plaća naknada od € 100 za svaku prijavljenu igru. Time se izbjegao i još jedan problem od Greenlighta gdje bi netko postavio desetke inačica jedne igre sa malo drugačijim naslovom i grafikama te tako „spamao“ sustav u nadi da bar jedna prođe jer nije bilo rizika kad je naknada bila jednokratna.

U periodu početkom 2017. kada je najavljeno gašenje Greenlighta postojala je velika neizvjesnost jer Valve nije još u to vrijeme odlučio kolika će biti naknada za prijavu svake igre i najavili su brojku između 100 i 5000 što je zabrinulo male studije jer bi u slučaju € 5000 to jako loše utjecalo na mogućnost nekih da nastave rad u toj industriji. Nedugo prije aktivacije Directa najavljena je naknada od € 100 na opće zadovoljstvo developera s obrazloženjem da nakon mnogo razmišljanja u tvrtci nisu mogli opravdati bilo koju višu cjenu. Uz to Valve je napomenuo i implementirao mogućnost povrata tih € 100 nakon ostvarenih € 1000 od prodaje u trgovini po lansiranju igre.[27]

Dok su s jedne strane developeri bili oduševljeni moglo bi se reći da su neki korisnici i kritičari bili zabrinuti jer je prag ulaska u trgovinu postao nizak što bi u kombinaciji s činjenicom da ne postoji gotovo nikakva kontrola kvalitete za igre koja prolaze kroz Direct osim provjere ispravnosti točnih navoda izdavača moglo preplaviti trgovinu nekvalitetnim igrama. Do dana pisanja ovog rada to se još nije dogodilo.[10]

7. Proces razvoja videoigre

Kod izrade videoigri potrebno je pronaći ljude sa vještinama u programiranju, grafičkom oblikovanju i animaciji, bilo to 3D ili 2D ovisi o zahtjevima projekta, UX, snimanju i obradi zvuka i glazbe. Uz to je poželjno i imati nekoga sa znanjem u području marketinga i odnosima s javnošću.

Kada se kreće u izradu videoigre ne postoje fiksirana pravila kojih se treba pridržavati i opseg svih aspekata u procesu se stalno mijenja i širi s porastom mogućnosti računala i novim programskim tehnologijama. Videoigra se prepoznaje po svojim sastavnim komponentama koje su najčešće sljedeće : vizualno oblikovanje, programski kod, sučelje, zvučni efekti, glazba i kontrole. Kod svakog pravila postoje iznimke pa tako i ovdje ima slučajeva gdje neke igre nemaju sve prethodno navedene komponente ili postoje proizvodi koji ih imaju ali se ne prepoznaju kao videoigre.

Svaka igra počinje od ideje. Izvedivost ideje se ispituje u prototipu koji se u industriji još i naziva demo u slučaju da se distribuira korisnicima za svrhe testiranja ili neke druge potrebe i obično je besplatan. Danas postoji termin „Early Access“ kojim se obilježavaju igre koje su u fazi razvoja ali se već prodaju. S jedne strane to je dobro za developere koji dobivaju profit tjekom izrade igre i mogu uložiti u bolji razvoj no češće je slučaj da takve igre nebudu nikad završene jer tim koji stoji iza razvoja igre dobije novce i izgubi interes za daljnjim razvojem jer je cilj zarade već postignut.

U prototipu se naprimjer rjetko uključuju zvučne komponente. Prototip se često naziva i srž (eng. core) iz razloga jer se u ovoj fazi razvija srž igre odnosno elemnti koji će biti pokretačka sila cjelog proizvoda i koji su primarni u razvoju i moraju funkcionirati u svakom slučaju, ostali elementi se oblikuju oko toga. Faze razvoja u industriji videoigara se djele na alpha, beta i final i pravila za obilježavanje nisu nigdje definirana pa je obilježavanje igre sa bilokojim od tih oznaka subjektivna procjena onoga koji ju razvija.

Srž igre može biti bilo koji element, bio to zvuk, grafika, kontrole ili logika. U svakom slučaju logika je ljepilo koje drži sve ostale elemente na okupu i u skladnom odnosu pa se sa izradom prototipa počinje od logike. U početku se mogu koristiti takozvani „placeholderi“ što bi se doslovno prevelo kao držaći mjesta. Placeholder 2D lika u igri može biti običan jednobojni kvadrat na kojeg se veže logika i kontrole. Kasnije kada se radi na vizualnim rješenjima na mjesto kvadrata se stavlja animacija lika i onda se rješava dio logike vezan za animaciju kad sve ostalo radi kako bi trebalo. UX uvjek treba uzimati u obzir jer se već kod biranja kontrola treba imati na umu korisničko iskustvo kod korištenja tih kontrola u konačnom proizvodu.

Kada su sve animacije finalne ili blizu finalnom kreće rad sa zvukom. Razlog tome je što se treba odrediti trajanje zvuka u igri prije nego se može odrediti kakav zvuk je potreban za svaki slučaj gdje ga se planira postaviti. Svaki game engine nudi opciju podešavanja zvuka pa se manje promjene mogu izvoditi i tamo.

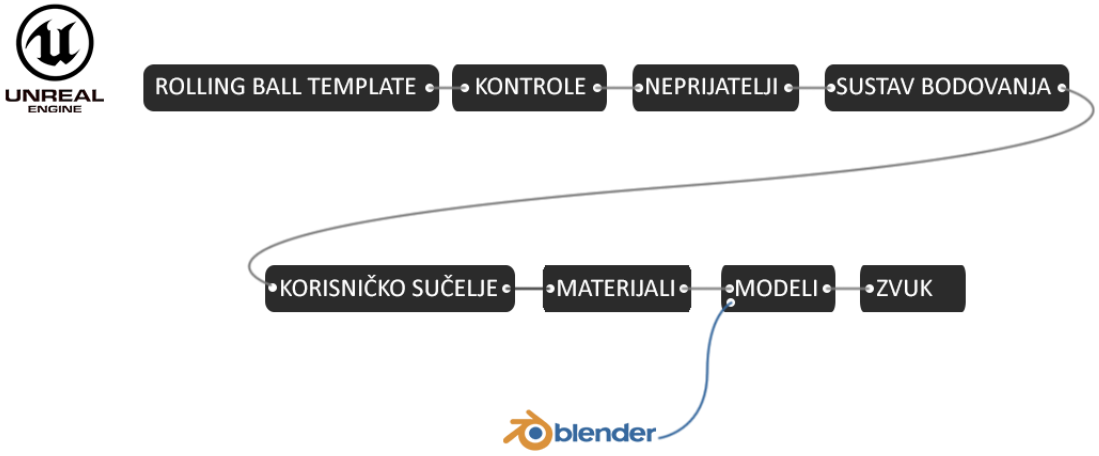
Od početka procesa razvoja videoigre uvjek je dobro vokalizirati ideje sa timom i staviti sve u neki fizički oblik, npr. na papir ili radnu ploču. Važno je imati viziju kako će elementi funkcionirati u konačnoj verziji kako bi se odpočetka moglo pripremati na nadolazeće elemente koji trebaju biti uključeni i izbjegla potreba za prerađivanjem velikih djelova logike.

Prije početka projekta dobro je i razmisliti o ciljanoj publici i pregledati tržište da se može procjeniti koliko rada ima smisla uložiti u projekt. Zbog kompleksnosti samog medija videoigre gotovo je nemoguće da se može pojaviti slučajno kloniranje pa je većina klonova videoigara namjerna jer se žele priključiti na slavu glavne igre koju kloniraju. To se ne odnosi na igre koje su slične samo po nekoj komponenti drugima jer su uvjek razvijene sa nekom posebnosti koja doprinosi njezinoj jedinstvenosti što je i slučaj kod drugih medija.

Nakon početka distribucije konačne verzije igre rad na njoj ne prestaje, kao što je to uvjek slučaj sa interaktivnim programima, jer se uvjek nađe nepredviđena situacija na koju se tjekom razvoja nije računalo a koja izađe na vidjelo zbog količine ljudi koji ju igraju. Neke greške su nerjetko toliko specifične da se provuku i kroz faze testiranja.

Na igri je rad gotov kada se to odluči, pogotovo u današnjem svjetu interneta gdje se pod određenim uvjetima ažuriranje može izvoditi gotovo trenutačno. Osim toga nerjetko je danas vidjeti i ekspanzije za videoigre koje su dodaci na osnovnu igru koja je već u prodaji bilo koji period vremena i ponekad mogu uvelike doprinjeti igri i oživjeti ju i nakon nekoliko godina.

Dijagram procesa razvoja igre u slučaju prezentiranog primjera.



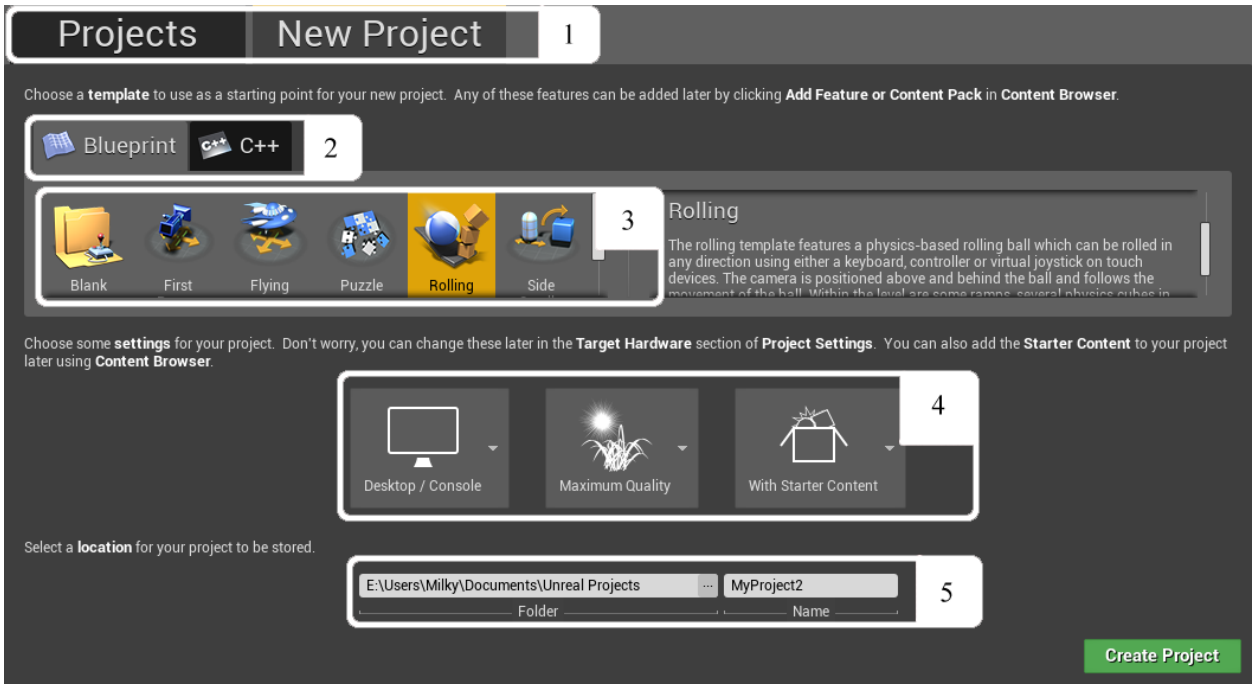
Slika 7.1 Proces izrade videoigre

8. Primjer razvoja igre

U radu će za sljedeće biti predstavljen primjer razvoja jednostavne 3D igre uz pomoć Blendera i UE4 kroz korake. Treba uzeti u obzir da autor rada ima tek par sati iskustva u radu sa UE4 kako bi se dobila perspektiva za svakoga tko se se želi okušati u tome kako bi mu se predočio opseg rada koji je moguće odraditi u predstavljenim okvirima. Primjer videoigre koja je predstavljena je izrađena u vremenskom okviru od 10 dana.

Idaja iza videoigre je kombinirati vrstu igre gdje igrač kontrolira kuglu, tzv. „roller“ sa klasičnom igrom Pac-man.

Po pokretanju Unreal Engine 4 pojavljuje se Unreal Project Browser početni zaslon koji nudi opcije za početak rada.

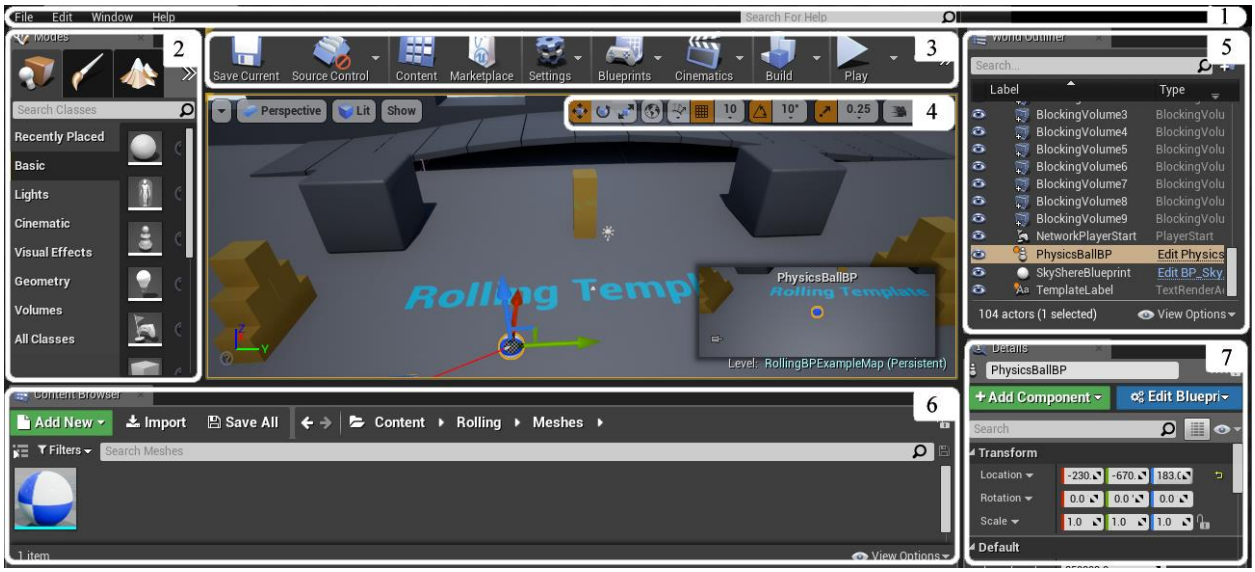


Slika 8.1 Unreal Project Browser

Unreal Project Browser ima nekoliko elemenata koji su za svrhu ovog rada istaknuti bjelim okvirima i numerirani za jednostavnije referenciranje. U okviru pod brojem 1 nalaze se 2 kartice. Kartica Projects nudi sve dosadašnje spremljene projekte za nastavak rada na njima dok kartica New Project nudi postavke za stvaranje novog projekta. Na karticama u okviru pod brojem 2 se bira hoće li projekt biti rađen u Blueprint sustavu ili sa C++. U okviru pod brojem 3 nalaze se svi predlošci koje Epic nudi za početak izrade igre sa grafikama koje predstavljaju tip predloška i s opisom predloška s desne strane. Okvir broj 4 nudi postavke za prilagođavanje platformi za koju je igra namjenjena, grafičke mogućnosti i besplatni sadržaj za početnike.

U poljima pod brojem 5 se navigira do mape u koju će se projekt spremati i upisuje naziv projekta. Za potreba ovog primjera postavke projekta su identične onima na slici. Nakon što se odaberu sve željene postavke pritisne se Zelena tipka u desnom kutu Create Project i ulazimo u Unreal Editor.

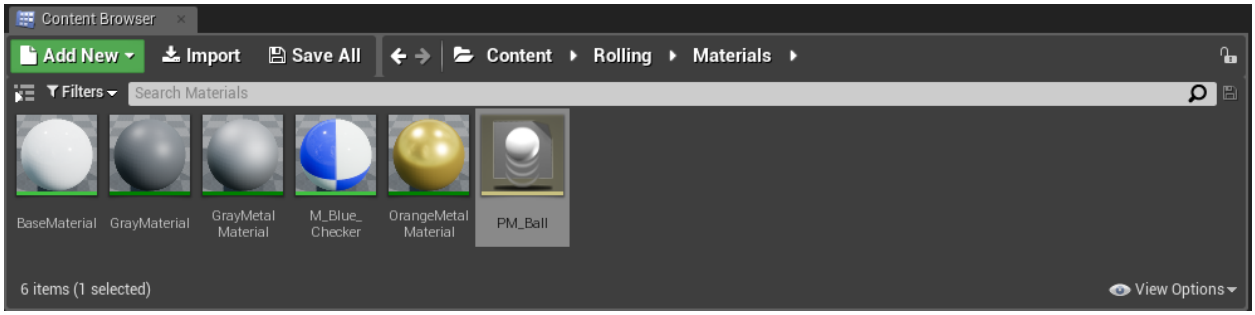
Unreal editor je glavno sučelje kroz koji se pristupa ostalim editorima i sastoji se od više panela od koji svaki ima različitu namjenu. Svaki panel se može premjestiti, zatvoriti i prediomenzionirati. Glavni prozor u oblikovanju projekta na kojem se radi naziva se Level editor [11] i on je temeljni element iz kojeg se stvaraju leveli, postavljaju i modificiraju elementi na sceni i vrši testiranje igre. U sredini se nalazi 3D pogled u scenu za odabrani level i samo jedan može biti odabran u danom trenutku. Desni gornji kut pogleda nudi opcije prikaza pogleda.



Slika 8.2 Unreal Editor

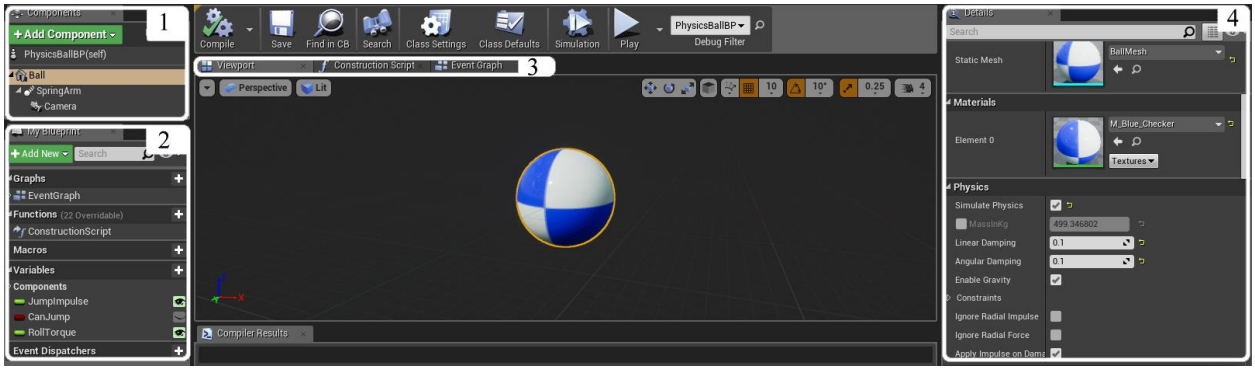
U odjeljku pod brojem 1 nalazi se Meni nalik na menije drugih Windows aplikacija koji se koristi za pristup generalnim postavkama projekta i samog editora. Broj 2 Modes nudi kartice s mogućnostima za izvršavanje specifičnih zadataka na levelu kao što su stvaranje novih elemenata na sceni, bojanje 3D oblika na sceni i modifikacija okoliša i terena. Alatna traka koja se nalazi pod brojem 3 pruža pristup često korištenim alatima i mogućnostima. Razne ikone pod brojem 4 predstavljaju alate za manipulaciju elementima na sceni nalik na alate za manipulaciju u drugim programima za 3D modeliranje. Pod brojem 5 se nalazi World Outliner, popis svih elemenata koji se nalaze na sceni slično layers panelu u Adobeovim programima kao što su Photoshop i Illustrator. U panelu broj 6 nalazi se preglednik sadržaja Content Browser kroz koji se pristupa i stvaraju novi elementi igre kao što su materijali, aktori i leveli. Ako se odabere bilokoji element u sceni u panelu Details pod brojem 7 prikazuju se njegove specifične pojedinosti.

Prvo treba vidjeti kako funkcioniraju stvari koje su uključene u projektu po predlošku. Igre su uvjek mogu testirati pritiskom na tipku Play u alatnoj traci. Po testiranju izmjenjene su neke postavke za bolju kontrolu kugle. Prvo je pristupljeno Physics Materialu pod nazivom PM_Ball. Physics Materiali su elementi koji se mogu dodati na elemente igre sa aktiviranim simulacijama fizike kako bi se postavile vrijednosti materijala u odnosu sa drugim materijalima kada se simulira fizika. U ovom slučaju materijal se nalazio pod Content/Rolling/Materials.



Slika 8.3 PM_Ball

Dobro je znati da je lokacija svih elemenata u izborniku sadržaja proizvoljna, a kategorizacija po mapama služi samo za bolju orijentaciju pri radu. Dvoklikom na PM_Ball otvaraju se opcije u novom prozoru. Po potrebi novo otvoreni prozori mogu se odvući iznad menija u novu karticu. U ovom slučaju povećana je vrijednost Friction s 0.8 na 3 kako bi se povečalo trenje i smanjila klizavost kugle, smanjen Density s 1.5 na 1 i Raise Mass to Power s 0.75 na 0.65 kako bi kugla bila lakša i responzivnija. Blueprint je element koji povezuje više elemenata u jednog aktora i nosi na sebi logiku.Blueprint od kugle, PhysicsBallBP se nalazi u Content/RollingBP/Blueprints. Nakon otvaranja otvra se prozor na slici.



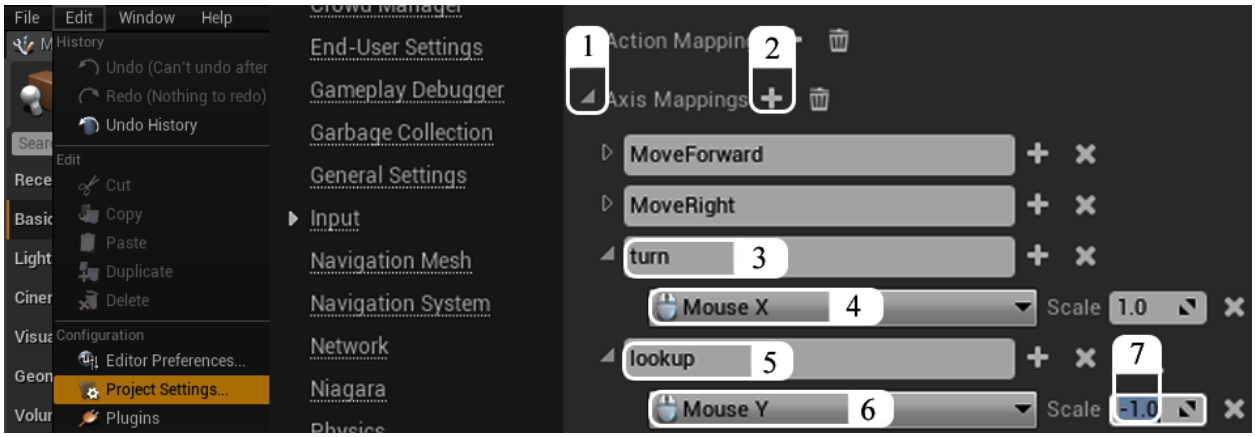
Slika 8.4 PhysicsBallBP

Svaki prozor blueprinta je podjeljen na nekoliko panela.od kojih su neki identični onima u Level editoru. U okviru 1 Components nalazi se panel sa komponentama koji se nalaze u ovom blueprintu.

U okviru broj 2 „My Blueprint“ nalaze se sve funkcije i varijable ovog blueprinta. Pod 3 se nalaze 3 različite kartice. Prva je Viewport u kojoj je 3D pogled gdje se vide svi elementi blueprinta kako bi izgledali kada bi ga se stavilo u scenu. Druga kartica je Construction Script. Tu su funkcije koje se izvršavaju prije Event Grapha i mogu služiti prilagođavanju Blueprinta različitim potrebama izmjenjivanjem vrijednosti i odnosa varijabli u Event Graphu prije njegovog pokretanja, npr u ovom slučaju ako bi se htjelo postići da nam kugla na jednom levelu skaće više ili kotrlja brže nego na drugom levelu to bi izmjenili ovdje. Treći panel je Event Graph i to je glavni panel u kojem se stvara glavna logika u Blueprint sustavu. U okviru broj 4 se nalazi Details panel u kojem se mogu vidjeti sve vrijednosti koje su podesive po odabiru neke komponente u prvom panelu.

Kako bi se još malo bolje prilagodilo kontroliranje kugle odabirom na Ball u listi komponenata i pogledom u Details tab u kategoriji Physics, uključi se opcija Max Angular Velocity i njena vrijednost se postavi na 1500. Time će se ograničiti pretjerano brzo okretanje kugle pri kontroliranju. Kako igra ne sadržava puno elementi svi će biti stavljeni u glavnu mapu Content kako bi bili na jednom mjestu, brzo dostupni i odvojeni od sadržaja koji je došao sa predloškom.

Kako bi se dodala kontrola nad kamerom moraju se u postavkama definirati vrjednosti za pomicanje miša. U meni traci na vrhu navigira se do Edit/Poject Settings/Input



Slika 8.5 Input Settings

U Bindings kategoriji prošire se opcije pod Axis Mappings pritiskom na trokutić označen brojem 1 na slici. Pritiskom dva puta na plus ikonu označenu brojem 2 dodaju se 2 nove kontrole. Jednu koja je u ovom slučaju nazvana „turn“ pod brojem 3 poveže se sa kontrolom Mouse X iz izbornika označenog brojem 4 ondosno X osi na mišu koja je lijevo-desno i ostavljena joj vrijednost 1, a drugu koja je nazvana „lookup“ i označena brojem 5 povezana s kontrolom Mouse Y iz izbornika pod brojem 6 odnosno pokreti miša gore-dolje i vrijednost označena brojem 7 je postavljena na -1.

Sada se u svakom blueprintu mogu referencirati nove kontrole. U ovom slučaju potrebno ih je povezati sa orijentacijom kamere koja je pod-komponenta PhysicsBallBP. Povratkom u PhysicsBallBP i odbirom kartice Event Graph dolazi se do sučelja za izradu logike. Novi elementi dodaju se pritiskom desne tipke miša na prazan prostor u matrici. Prilikom toga otvara se novi izbornik koji nudi elemente koji se mogu koristiti za izradu logike. U prozoru je polje za pisanje kojim se lakše mogu pronaći elementi ako se zna njihov naziv. Ako je uključena opcija Context Sensitive izbornik će prikazivati samo one elemente koji imaju logičkog smisla da budu korišteni u danom kontekstu.

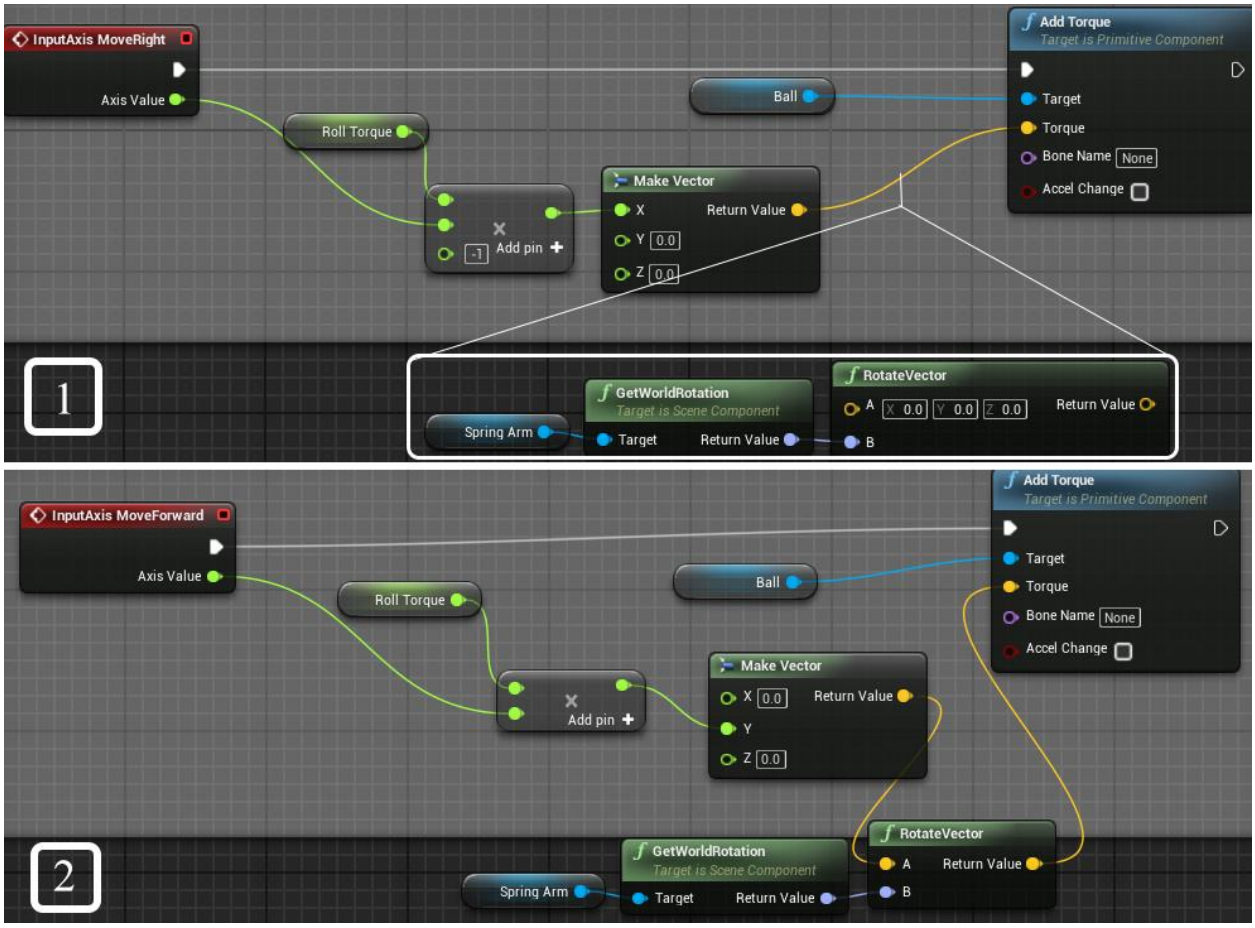
Slika 8.6 Dodavanje elemenata

Uz pomoć pretraživanja dodaje se prethodno stvorena kontrola turn. Tip elementa kontrole je takav da ima samo izlazna sidrišta jer je on inicijator. To znači da će se, kada računalo registrira unos iz te kontrole, izvršiti logika koja proizlazi iz tog elementa. Elementi se spajaju u linearni sljed linijama povlačenjem miša iz jednog sidrišta u drugo. Za odvajanje linije od sidrišta drži se tipka Alt i pritisne lijeva tipka miša. Elementi se mogu dodati i na način da se iz jednog sidrišta povuče miš u prazni prostor prilikom čega će se otvoriti izbornik. U ovom slučaju potrebno je dodati Add Controller Yaw Input koji se pomoću pretraživanja brzo pronalazi upisom rječi „yaw“. Ovaj postupak se treba ponoviti i za drugu prethodno stvorenu kontrolu lookup ali sa razlikom da se umjesto elementa Add Controller Yaw Input postavlja element Add Controller Pitch Input. Nakon toga se Axis Value spaja na Val za svaki par elemenata. Po završetku ovoga logika izgleda koa na slici.

Slika 8.7 Kontrola kamere

Važno je napomenuti da razmještaj elemenata ne utječe na njihov rad već služi samo za vizualnu orijentaciju. Jedino je važno koji elementi su spojeni kroz sidrišta i u kojem redosljedu. Redosljed počinje sa prvim trokutastim sidrištem u svakom elementu i nastavlja se kroz ta trokutasta sidrišta. U primjeru na slici postoje dvije početne točke jedna u Input Axis lookup a druga u Input Axis turn i one funkcioniraju neovisno jedna o drugoj.

Nakon što su uspješno dodane kontrole u blueprint i sidrišta su spojena kao i na slici neovisno o rasporedu po matrici treba doraditi logiku za kretanje koja je došla sa predloškom. U zakomentiranoj površini pod nazivom „Roll left/right“ i „Roll forwards/backwards“ dodaju se novi elementi na način demonstriran u prethodnom koraku. Elementi su GetWorldRotation (Spring Arm) koji nosi vrijednosr rotacije nosača kamere relativno na svijet i element RotateVector koji zaokreće dani vektor u sidrište A za vrijednost rotacije danu u sidruštu B. GetWorldRotation će biti vrijednost B a Return Value u Make Vector elementu će biti vrijednost A. Return Value u elementu Rotate Vector će biti uključena u Torque u elementu Add Torque. Radi lakše predođbe u slici je prikazano u dva koraka kako se to izvodi.



Slika 8.8 Modifikacija logike iz predloška

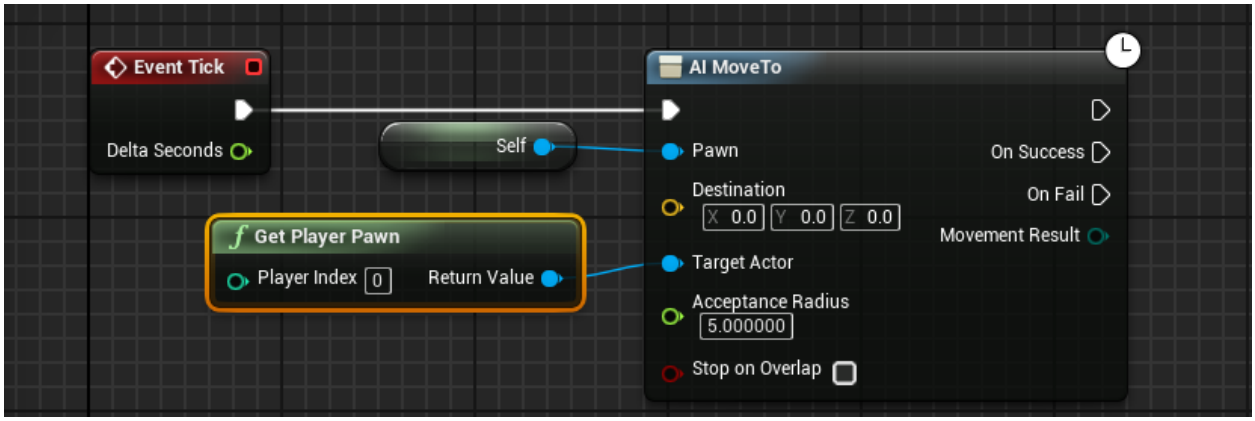
Za kraj ovog djela odabere se SpringArm u Components panelu. U Details panelu uključe se opcije Do Collision Test pod kategorijom Camera Collision i opcija Use Pawn Controller pod kategorijom Camera Settings. Do Collision Test spriječava prodiranje kamere kroz druge elemente na sceni a Use Pawn Controller referencira orijentaciju kugle kako bi se nosač kamere mogao okretati relativno na nju. Po završetku pritiskom na tipku Compile u alatnoj traci kompilira se nova logika i trenutna verzja igre se može iztestirati pritiskom na tipku Play u alatnoj traci kako bi se provjerila ispravnost kontrole nad kamerom.

Kada je upravljanje kugle rješeno može krenuti dodavanje neprijatelja. Redosljed dodavanja elemenata nije bitan osim u slučaju da se neki element referencira na neki drugi. Naprimjer nebi bilo moguće napraviti neprijatelja koji prati igrača ako još nije stvoren blueprint igrača na koji će se nepritelj referencirati kako bi dobio lokaciju prema kojoj se treba kretati.

Svi novi elementi igre su u ovom primjeru stavljeni u glavni direkorij Content kako se nebi mješali sa elementima iz predloška i zbog lakše navigacije. U ovoj igri neće biti mnogo elemenata pa nije potrebno razvrstavati ih u mape.

Prije nego se može raditi logika sa AI navigacijom mora se u sceni postaviti Nav Mesh Bounds Volume koji se nalazi u panelu Modes i služi generiranju puteva kojim se AI može kretati s obzirom na prepreke u sceni. Prikaz generiranih puteva u pogledu na scenu može se aktivirati pritiskom tipke P nakon selekcije Nav Mesh Bounds Volume kocke. Putevi će se generirati samo na elementima scene unutar kocke.

Desnim klikom miša na prazno polje u Content Browseru i navigacijom do Blueprints/BlueprintClass otvara se izbornik tipova blueprinta koji se mogu stvoriti. Za ovaj slučaj je odabran Character. Character je Pawn koji ima mogućnosti kretanja sa kontrolama dok je Pawn Actor koji ima mogućnost da igrač sa njim upravlja. U primjeru je novo stvoreni blueprint nazvan „Neprijatelj1“. Dvoklikom na Neprijatelj1 otvara se njegov blueprint. U Components panelu je pritiskom na tipku Add Component dodaje se komponenta Cube pod kategorijom Common. Kvadrat služi kao placeholder kako bi bilo vidljivo kud se neprijatelj kreće. U Event Graphu Neprijatelja1 dodaje se logika za praćenje igrača.

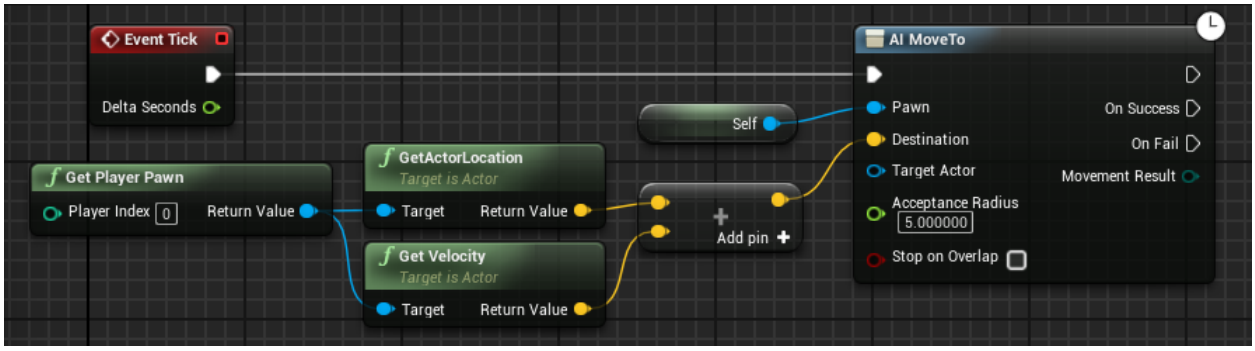


Slika 8.9 Logika Neprijatelja1

Element Event Tick je početni element koji se izvršava svaku sliku u sekundi (eng. FPS). Ako igra radi na 60 FPSa znači da se u sekundi prikaže 60 sličica na ekranu, a element Tick se izvrši 60 puta. AI MoveTo je element koji iza sebe ima logiku kojom navigira kroz Nav Mesh Bounds Volume. Pawn je element koji će se kretati pod utjecajem AI MoveTo elementa a Target Actor je cilj prema kojem će se Pawn nastojati kretati.

Blueprint Neprijatelj1 se postavlja u scenu povlačanjem njega samog mišom iz Content Browsera u scenu. Pritiskom na Play može se testirati kako sve funkcionira u ovoj fazi

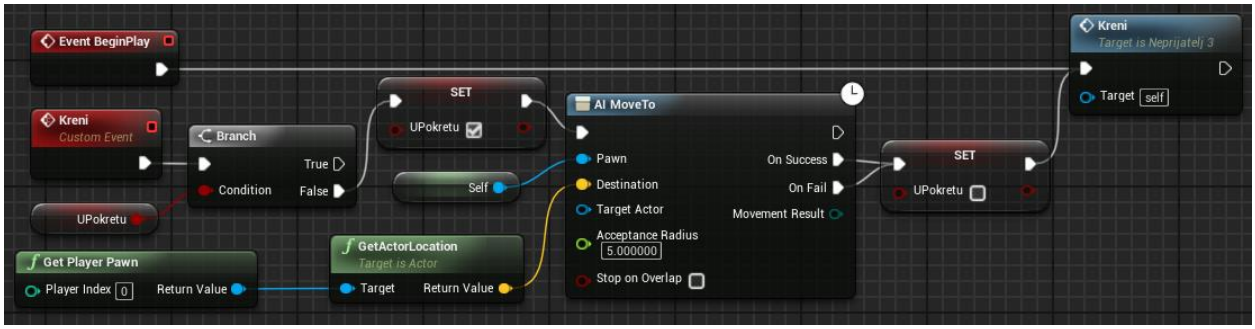
Sljedeće se dodaju još 3 različita neprijatelja. Svaki od neprijatelja treba imati drugačiju logiku kretanja kao što je to i u originalnom Pac-manu. Da bi se dodalo još neprijatelja blueprint Neprijatelj1 se duplicira desnim klikom na njega u Content Browseru i odabirom opcije Duplicate. UE će automatski nazvati novi blueprint Neprijatelj2. Otvaranjem blueprinta Neprijatelj2 i njegovog Event Grapha izmjenjuju se neki elementi u logici kako bi se Neprijatelj2 kreto drugačije.



Slika 8.10 Logika Neprijatelja2

Logika za blueprint Neprijatelj2 je drugačija od Neprijatelj1 po tome što se ovdje izvlači lokacija iz igrača pomoću elementa Get Actor Location koji se kroz Target referencira na igrača i Get Velocity koji iz igrača izvlači njegovu brzinu. Zbrajanjem ta dva vektora dobije se vrh vektora koji je uvjek ispred igrača relativno na smjer njegova kretanja i udaljen od njega dalje čim je brzina veća. Takva logika upravlja Neprijateljom2 tako da će on uvjek nastojati presresti put igraču. Neprijatelj se može postaviti u scenu i igra testirati.

Ponovnim dupliciranjem bilo blueprinta Neprijatelj1 ili Neprijatelj2 dobiva se blueprint Neprijatelj3. Za Naprijatelja3 je napravljena logika po kojoj se kreće do lokacije igrača u tom renutku i tek po dolasku na tu lokaciju ponovo pribavlja lokaciju igrača i kreće se prema sljedećoj. Logika za Neprijatelja3 izgleda kao na slici.



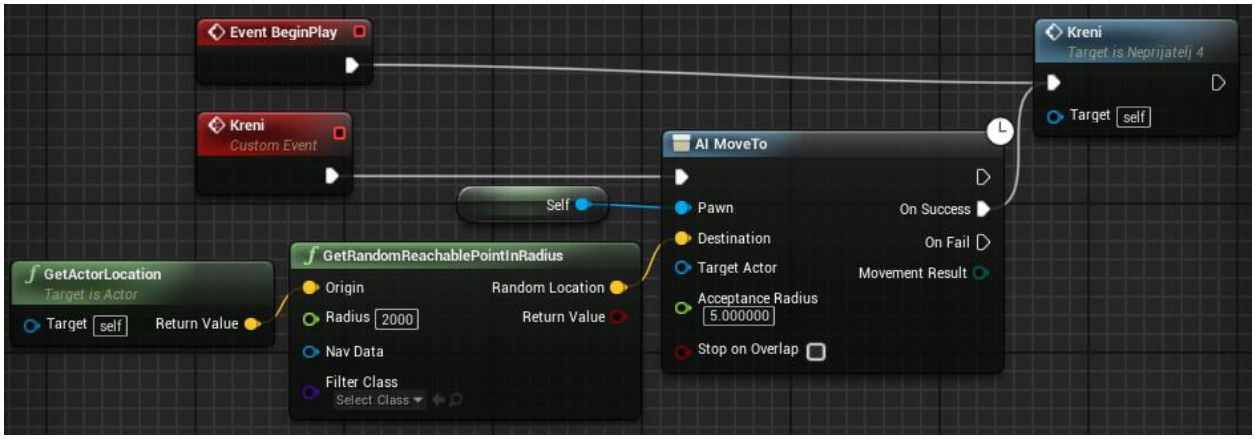
Slika 8.11 Logika Neprijatelja3

Za ovaj slučaj trebao se izraditi Custom Event. Taj element se pronalazi u izborniku kada se pri desnom kliku na prazni prostor otvore svi elementi i upiše riječ „custom“. Custom Event se može nazvati po želji, a u ovom slučaju nazvan je „Kreni“. Potrebna je i jedna varijabla tipa Boolean koja će provjeravati ako se Neprijatelj3 kreće. U slučaju da se kreće on još nije stigao na odredište i neće se pribaviti novo odredište dok ne stigne. Nove varijable se stvaraju u panelu My Blueprint s lijeve strane ispod Components panela pod kategorijom Variables pritiskom na ikonu „+“ Tip varijable se određuje odabirom varijable i postavkom Variable Type u Details panelu. Varijabla je nazvana UPokretu i po defaultu vrijednost joj je FALSE. Boolean varijable se mogu u Event Graph uvesti povlačenjem iz popisa varijabli prilikom čega se može uvesti u 2 oblika, Get ili Set. Get je element koji pribavlja trenutnu vrijednost varijable kao što je to na slici 11 spojeno u Condition sidrište Branch elementa a Set postavlja vrijednost varijable kao što je element povezan na False sidrište Branch elementa. Branch element je svojevrsni „if“ u programskim jezicima koji u ovom slučaju prima uvjet odnosno Condition koji može biti TRUE ili FALSE. Na izlazu su oba ishoda i bira se u kojem slučaju će se sljed logike nastaviti. U ovome je to False.

Umjesto incijatora Event Tick sada se za incijatora postavlja Custom Event Kreni. Custom event Kreni se pokreće po potrebi. Za početak treba ga bar jednom pokrenuti pa će se spojiti sa elementom Event Begin Play koji se pokreće samo jednom na početku igre.

U laičkim terminima ova logika radi ovako: Pri početku igre pokreni Custom Event Kreni. Kada je Kreni pokrenut provjeri u Branchu dali je UPokretu TRUE ili FALSE. Ako je FALSE pribavi lokaciju igrača u tom trenutku i pošalji Neprijatelja3 na tu lokaciju. Kada je Neprijatelj3 stigao na lokaciju (Success) ili nije uspio pribaviti lokaciju (Fail) postavi UPokretu na FALSE i ponovno pokreni Custom Event Kreni.

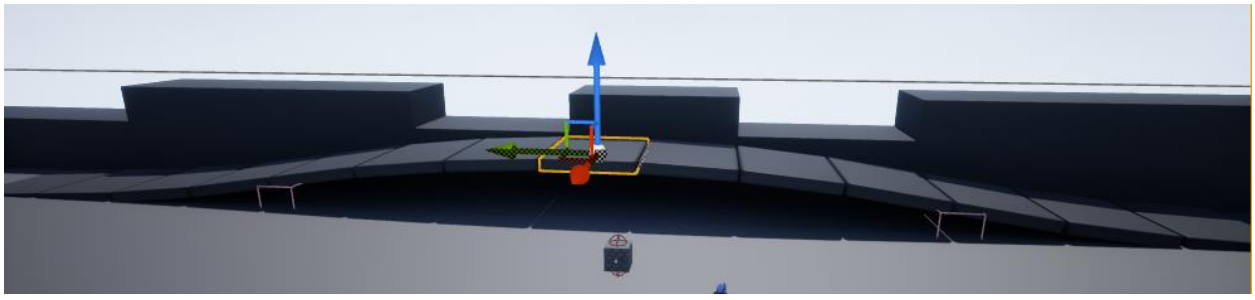
Posljednji neprijatelj, Neprijatelj4, imat će logiku po kojoj će samo nasumično lutati scenom.



Slika 8.12 Logika Neprijatelja4.

Kopiranjem blueprinta Neprijatelj3 može se iskoristiti Custom Event Kreni koji se isto zove kao i kod Neprijatelja3 ali nije povezan jer je svaki u svojem blueprintu. Kod Neprijatelja4 izbacimo Branch i sve elemente sa varijablom UPokretu i dodamo element Get Random Reachable Point In Radius u kojemo postavimo vrijednost Radius na 2000. Taj element uzima nasumičnu točku u krugu od 2000 jedinica oko trenutne lokacije Neprijatelja4 i AI Move To ga pošalje na tu lokaciju. Kada stigne na lokaciju uključuje se On Success i ponovo pokreće logika ispočetka.

Kada svi neprijatelji funkcioniraju kako je zamišljeno kreće sa stvaranjem sustava bodovanja. Sustav bodovanja je zamišljen na način da se iz Nav Mesh Bounds Volumea dobi nasumična točka i na njoj stvori bod. Kada igrač pokupi bod stvorit će se novi bod na drugoj nasumično odabranoj točci u sceni. Prije toga odlučeno je iz scene maknuti dvije rampe vidljive na slici 13. zbog nespretnog kretanja po njima.



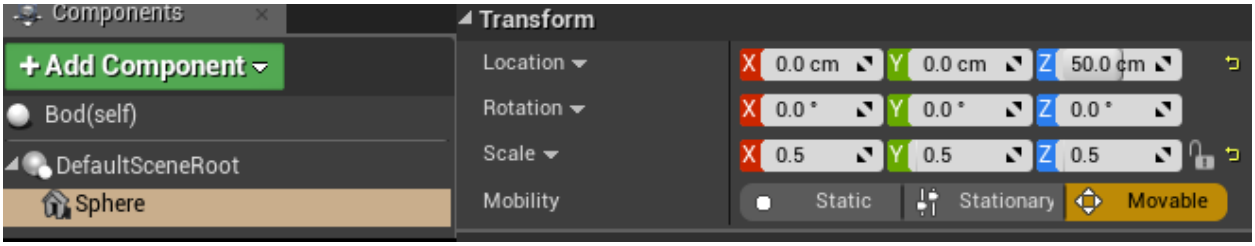
Slika 8.13 Problematične rampe

Umjesto rampi iz panela Modes, pod prvom karticom u kategoriji Basic na scenu su odvučena dva stožca, (eng. Cone). Uz pomoć alata u gornjem desnom kutu 3D pogleda stožci su uvećani i spljošteni te pozicionirani kao na slici.

Slika 8.14 Stožci

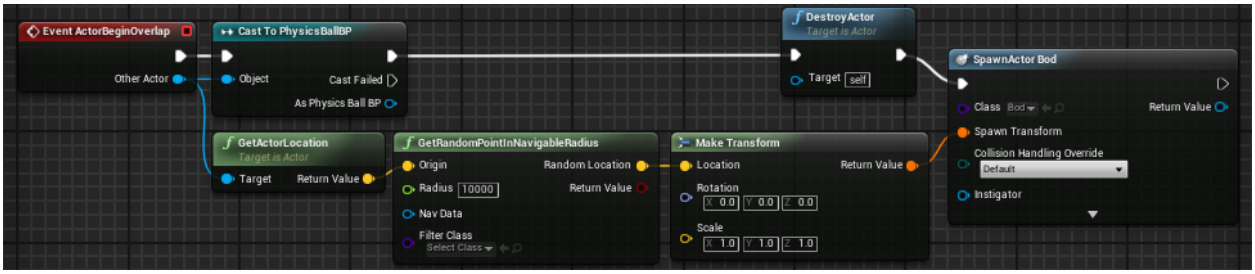
Osvjetljenje na ceni nije dinamično i mora se ručno izračunavati po potrebi da bi se uštedjeli resursi i lakše modificirali elementi scene. Svjetlo se ponovo izračuna odabirom Bluild/Build Lighting Only u alatnoj traci kraj tipke Play.

Kada je scena gotova izradi se novi blueprint klase Actor. U ovom slučaju nazvan je Bod. U blueprintu Bod dodaje se komponenta sfere u Components panelu i u Details panelu umanjuje na skalu od 0.5 i podiže 50 cm od izvorne točke kako pri stvaranju nebi bila ugrađena u pod. Također u Details panelu pod kategorijom Collision, Collision Presets se postavlja na OverlapAll kako se kugla nebi sudarila s Bodom ali bi i dalje registrirala koliziju.



Slika 8.15 Bod oblikovanje

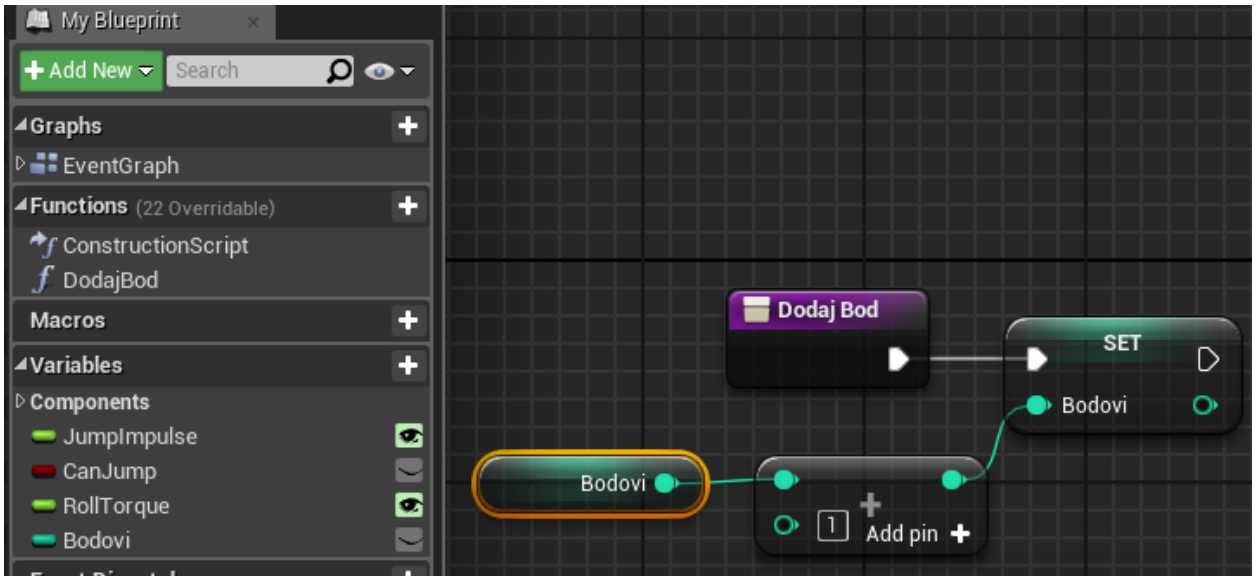
Logika za bod može se vidjeti na slici 8.16.



Slika 8.16 Logika Boda

Kada je sve postavljeno kao na slikama jedna instanca blueprinta Bod se postavlja u scenu.

U blueprintu PhysicsBallBP se u MyBlueprint panelu dodaje nova varijabla Bodovi tipa Integer i nova funkcija nazvana DodajBod. U novoj kartici Dodaj Bod stvara se logika koja dodaje vrijednost od 1 postojećoj vrijednosti u varijabli Bodovi.



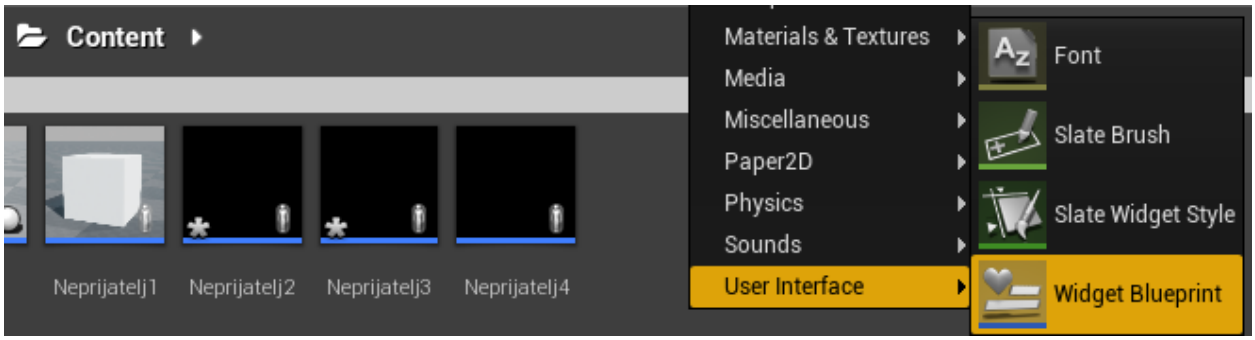
Slika 8.17 Funkcija Dodaj Bod

Razlog zbog čega se ova logika stvara u blueprintu PhysicsBallBP umjesto u blueprintu Bod je taj što je kugla konzistentno bez prekida na sceni dok se bod svaki put uništi i ponovo stvori pa bi u slučaju da je logika u blueprintu Bod vrijednost Bodovi svaki put resetirala na 0.

Za završetak logike u blueprintu Bod dodaje se element koji poziva u željenom trenutku funkciju Dodaj Bod. Funkciju Dodaj Bod jedino je moguće smjestiti nakon Cast to PhysicsBallBP jer taj element služi za pozivanje elemenata iz drugih blueprinta.

Slika 8.18 Pozivanje funkcije Dodaj Bod

Nakon završavanja logike treba postaviti sučelje koje će prikazivati stanje bodova na ekranu. Sučelja se u UE stvaraju sa Widget Blueprintom pod kategorijom User Interface.

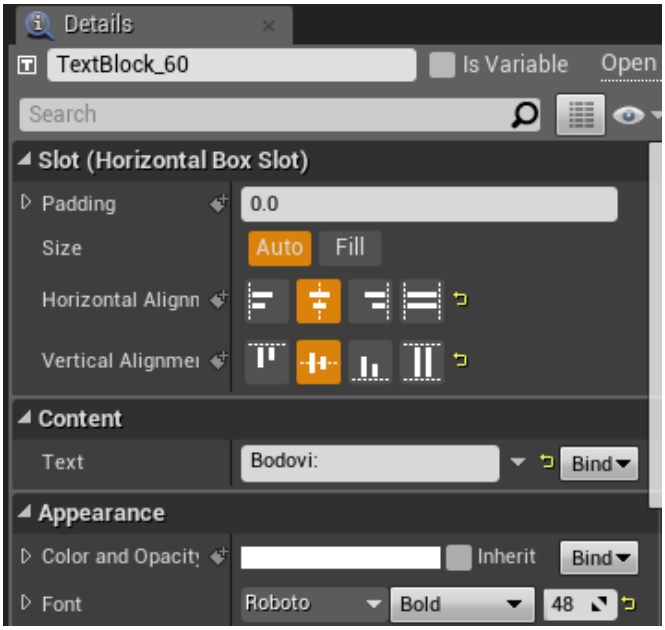


Slika 8.19 Widget Blueprint

Dvoklikom na novo stvoreni Widget otvara se njegov Blueprint. S lijeve strane nalazi se panel Palette u kojem je izbornik sa svim elementima korisničkog sučelja, a ispod njega je panel Hierarchy u kojemu su dodani elementi raspoređeni u hierarhiju isto kao što je to slučaj kod panela Components u prijašnjim blueprintima.

U panelu Palette se pronalazi Horizontal Box element i odvlači se u matricu u ljevi gornji kut zelenog okvira koji predstavlja prostor na ekranu. Ikonu nalik na cvjet koja predstavlja izvornu točku elementa odvuče se u sredinu kvadrata.

Iz palete se tada izvuče element Text i stavi unutar Horizontal Box elementa. U Details panelu za element Text postavi se Centrirano Horizontalno i Vertikalno poravnanje. Pod Content vrijednost polja text se postavi na „Bodovi:“ a pod Appearance veličina Fonta na 48.

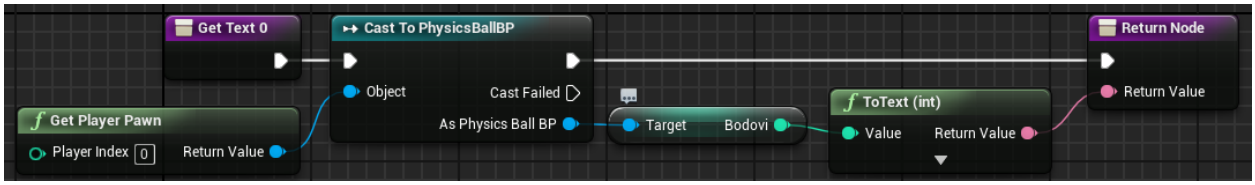


Slika 8.20 Text Bodovi

U Horizontal Box se smjesti još jedan Text element sa vrijednosti 999 i uključi se Horizontalno i Vertikalno poravnanje kao na prethodno. Nakon toga napravi se Bind.

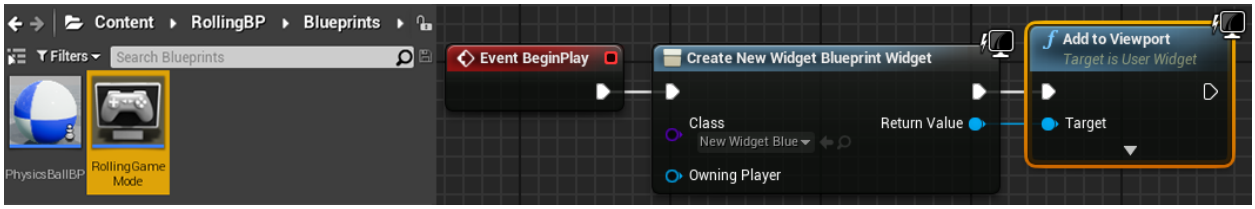
Slika 8.21 Bind

Pritiskom na Create Binding otvara se nova kartica gdje se može stvarati logika kao i u svakom drugom blueprintu. Početni elementi su Get Text 0 i Return Node. Između se dodaju elementi koji će povući vrijednost varijable Bodovi iz PhysicsBallBP blueprinta i konvertirati dobivenu vrijednost tipa Integer u vrijednost tipa Text. Na kraju dobivena logika izgleda kao na slici 22.



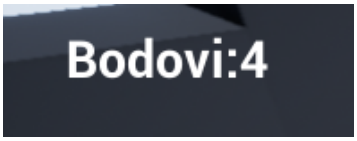
Slika 8.22 Get Text

Kako bi se dodao novo stvoreni Widget na ekran, mora se napraviti logika koja će izvršiti tu operaciju na početku igre. To se izvodi u RollingGameMode blueprintu koji nosi informacije o igri kao što je blueprint kojega igrač posjeduje i s kojim upravlja. Ovaj blueprint se nalazi u Content/RollingBP. Unutar njega doda se element Event Begin Play nakon kojeg se treba izvršiti element Create Widget. U njemu se iz izbornika Class odabere prethodno stvoreno Widget. taj element se veže na element Add to Viewport koji postavlja elemente na ekran, odnosno pogled.



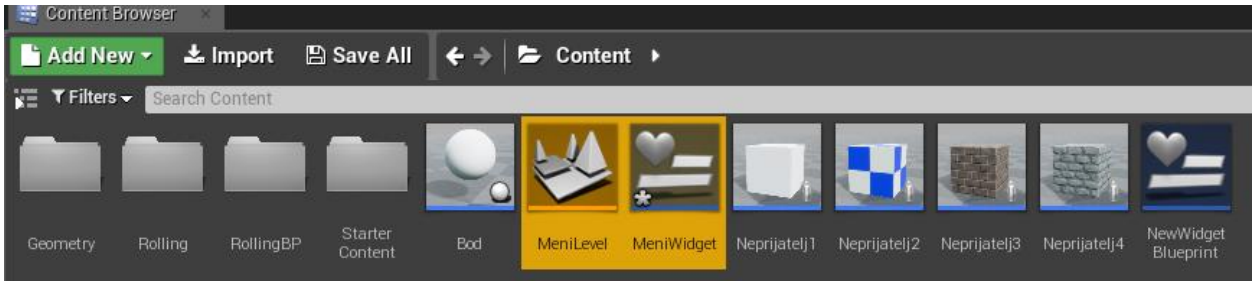
Slika 8.23 Add to Viewport

Kada su svi blueprinti kompilirani može se testirati ispravnost rada novih elemenata.
Ako je sve ispravno napravljeno na sučelju bi se trebalo prikazivati stanje bodova kao na slici.



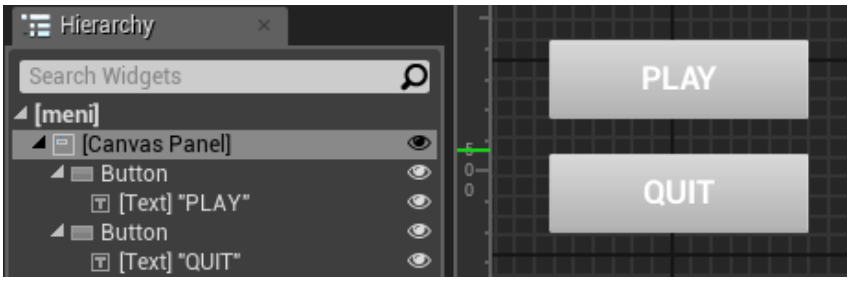
Slika 8.24 Prikaz bodova na sučelju

Sljedeće se dodaje glavni meni. U Content Browseru je stvoren novi Level i novi Widget. Level je nazvan MeniLevel a widget MeniWidget.



Slika 8.25 MeniLevel i MeniWidget

MeniWidget se otvori i u njega se postave dva Button elementa a u svaki od njih po jedan Text element. Jedan Text element se preimenuje u Play a drugi u Quit. Hierarhija treba izgledati kao na slici a pozicija i dimenzije gumbi je proizvoljna.

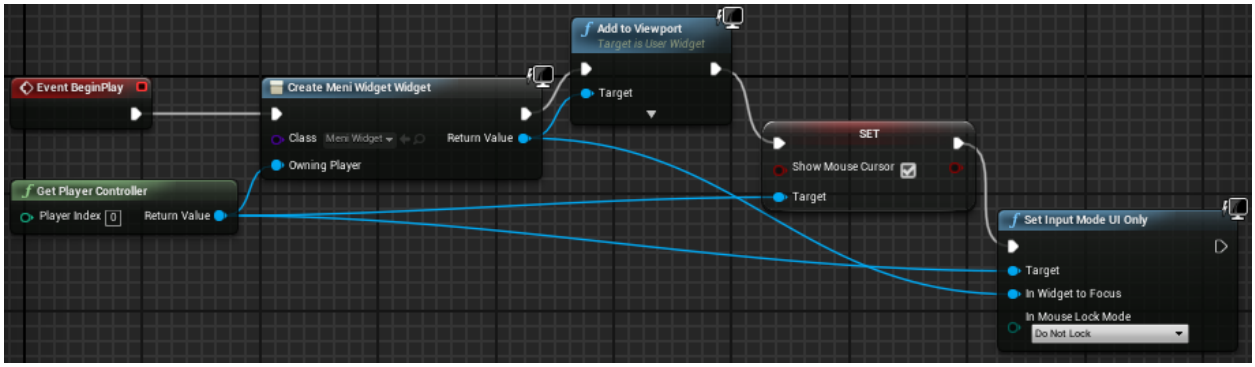


Slika 8.26 MeniWidget

Odabirom na prvi Button Play u Details panelu pod kategorijom Events treba dodati novi Event OnClicked pritiskom na zelenu tipku sa ikonom „+“. U novootvorenom Event Graphu postavlja se logika koja prilikom pritiska tipke Play otvara glavni level u kojem se odvija igra. Nakon toga pritiskom na tipku Desgner u gornjem lijevom kutu vraća se u Designer pogled. Odabirom drugog Buttona Quit i stvaranjem novog OnClicked Eventa za njega, ponovo se vraća u Event Graph i tamo slaže logika koja klikom na tipku gasi igru.

Slika 8.27 OnClicked Eventi

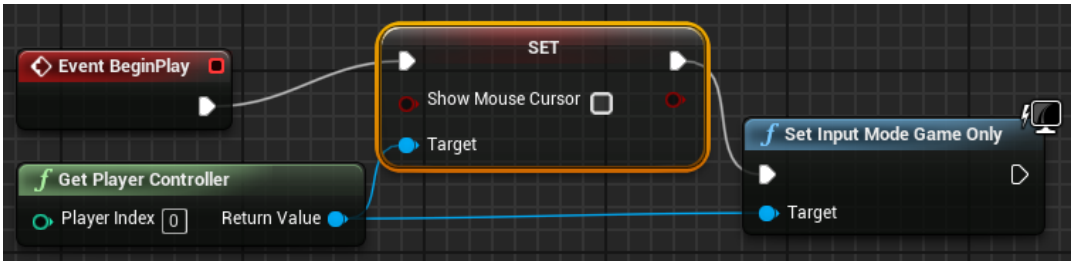
U Content Browseru se prebacuje na level MeniLevel dvoklikom na njega. Blueprint od levela se otvara opcijom Open Level Blueprint pod tipkom Blueprints u alatnoj traci. Potrebno je napraviti logiku koja će na sučelje zaljepiti tipke prilikom pokretanja levela MeniLevel kao što je to učinjeno sa bodovima. K tome potrebno je uključiti kontrolu mišom na MeniLevelu.



Slika 8.28 MeniLevel Blueprint

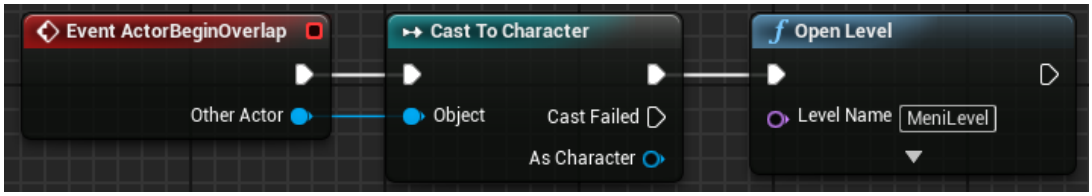
Povratkom na level RollingBPExampleMap otvara se njegov blueprint na isti način kao i kod MeniLevela, odabirom opcije Open Level Blueprint pod Blueprints tipkom u alatnoj traci.

Tamo se izradi logika koja pri pokretanju gasi kontrolu mišom i prebacuje ju na kontrole za igru. Pri pronalaženju elementa Set Show Mouse Cursor treba u ovom slučaju isključiti Context Sensitive opciju.



Slika 8.29 Set Input Mode

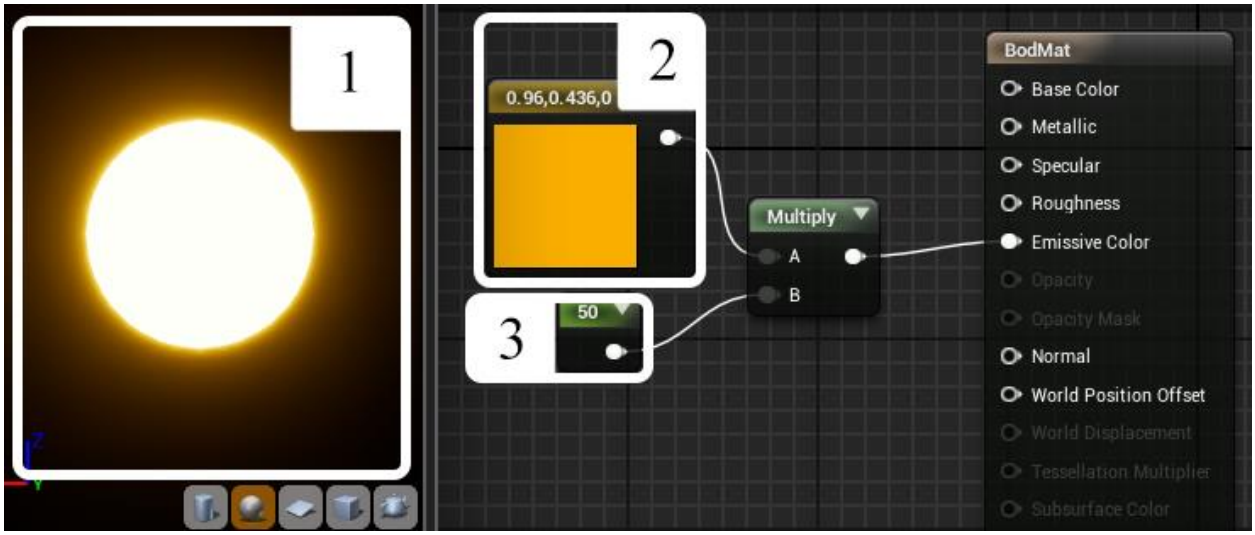
Sljedeći korak je postaviti osjetljivost na koliziju između igrača i neprijatelja. U slučaju da se igrač sudari s neprijateljom igra se treba prebaciti natrag u MeniLevel pri čemu se bodovi resetiraju. Logika se radi u PhysicsBallBP blueprintu i vrlo je jednostavna. Budući da su svi neprijatelji klase Character osjetljivost je postavljena na tu klasu kako se nebi trebala postavljati za svakog neprijatelja posebno. Uz to za komponentu Cube u Components kod svakog neprijatelja Collision Presets treba postaviti na OverlapAll kao što je to bilo sa blueprintom Bod.



Slika 8.30 Character Overlap

Nakon testiranja utvrđeno je da se pojavljuje greška kod stvaranja bodova koji se mogu stvoriti unutar zlatnih kutija sa simuliranom fizikom pa je kutijama smanjena masa promjenom vrijednosti Mass Sclae u Physics kategoriji u Details panelu s 1 na 0.05.

Ovime se završava faza mehanike i može se krenuti na vizualno oblikovanje igre. Vizualno se moglo oblikovati i prije ali ovime se izbjegavaju problemi koji bi se pri tomu mogli pojaviti. Prvo se kreće od najjednostavnijih stvari a u ovom slučaju to je dodavanje novih materijala.
Novi materijal se izradi desnim klikom na prazan prostor u Content Browseru. U ovome slučaju novi materijal je nazvan BodMat.



Slika 8.31 Bod Materijal

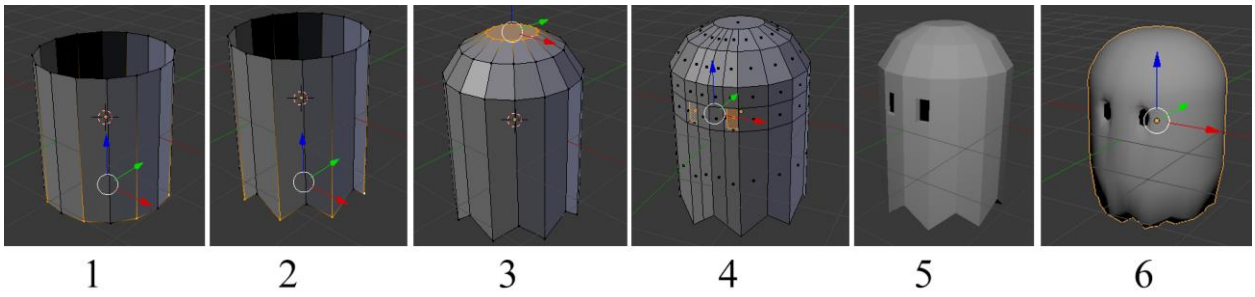
Materiali se modificiraju na isti princip blueprinta. U otkviru broj 1 može se vidjeti prikaz materijala u stvarnom vremenu. U grafu je glavni element BodMat u koji se priključuju vrijednosti materijala. U ovom slučaju bilo je potrebno napraviti materijal koji isijava boju. Dodana su dva nova elementa. Constant3Vector u okvoru broj 2 i Constant u okviru broj 3. Klikom na prvi element otvara se color picker i u ovom slučaju postavljena je žuta boja. Klikom na drugi element u Details panelu dodaje mu se vrijednost 50. Grafu se dodaje još jedan element Multiply koji množi unešene vrijednosti A i B i ta vrijednost se unosi u Emissive Color.

Kako bi se postavio materijal na blueprint Bod moramo otvoriti taj blueprint i odabrati Sphere u Components panelu. U Details panelu se pod kategorijom Materials nalazi izbornik iz kojeg se odabire materijal BodMat čime Sphere poprima prethodno izrađeni materijal.

Budući da je koncept igre baziran na Pac-manu odlučeno je obojati kuglu igrača u žutu boju kao što je to u klasičnoj igri. Zbog jednostavnosti i mogućih komplikacija odlučeno je izostaviti usta. U Content Browseru stvara se novi materijal BallMat i u grafu se samo priključuje Constant3Vector žute boje u Base Color. Nakon toga otvara se PhysicsBallBP i na isti način kao za bod odabere se Ball u Components panelu i promjeni materijal u Details panelu pod kategorijom Materials u prethodno stvoren BallMat.

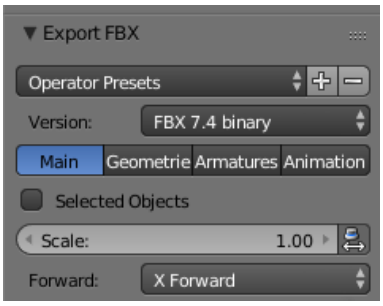
U ovoj fazi odlučeno je napraviti modele duhova u Blenderu za neprijatelje. Svi neprijatelji će djeliti jedan model ali svaki će materijal biti različite boje kako bi ih se razlikovali.

U 6 jednsotavnih koraka izrađuje se model duha u Blenderu.



Slika 8.32 Modeliranje duha

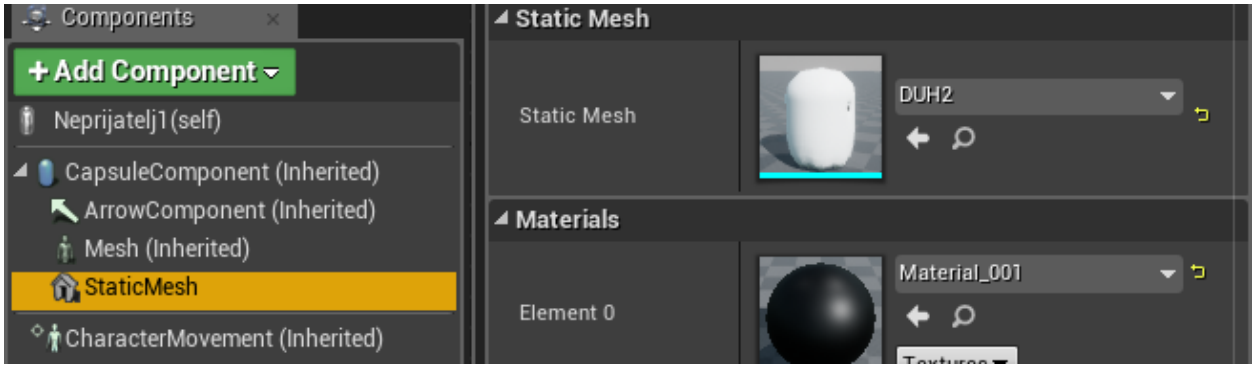
U prvom koraku dodaje se cilindar tipkama shift+a. Nakon toga selektira se svaki drugi vertex na donjem obruču i povuće prema dolje tipkom G. U trećem koraku odabire se gornji obruč i tipkom E izvlači i tipkom S skalira i taj postupak ponovlja još jednom. Na vrhu se tipkom F zapuni rupa. U koraku broj 4 dodaju se 2 obruća alatom Loop Cut u Transform alatima i odabriru dvije plohe za oči te se sa E ekstrudane i sa G odvlače unutra. U koraku 5 dodana su dva materijala, jedan na plohe oči a drugi na ostatak tjela. Boje su nebitne. Na posljetku je na cjeli model dodan Modifier Subdivison surface. Prije exportanja model se rotira za 90 stupnjeva po Z osi. Model se exporta u .fbx sa File/Export/FBX i u postavkama promjeni Forward u X Forward.



Slika 8.33 Export FBX

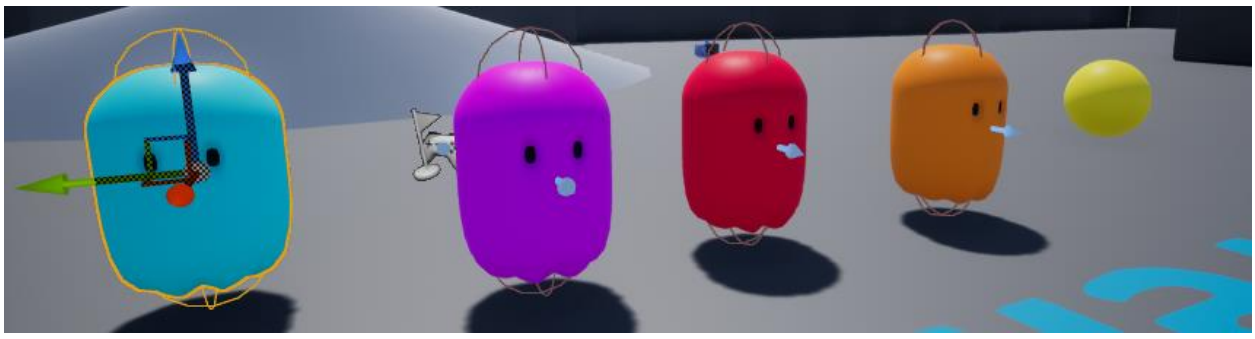
Povratkom u UE modele se može uvesti tipkom Import u Content Browseru. Odabirom FBX datoteke koja je prethodno exportana otvaraju se opcije za import. U ovom slučaju nisu mjenjane postavke već je odabrano Import All. UE automatski prepoznaje da je tip novog elementa Static Mesh i kao takvog ga postavlja.

Kako bi se zamjenile kocke koje su do tog trenutka predstavljale neprijatelje mora se u svakom blueprintu neprijatelja obrisati komponenta Cube i dodati nova komponenta Static Mesh. Odabirom te komponente u Details tabu pod kategorijom static Mesh odabire se prethodno importani Static Mesh duha.



Slika 8.34 Static Mesh

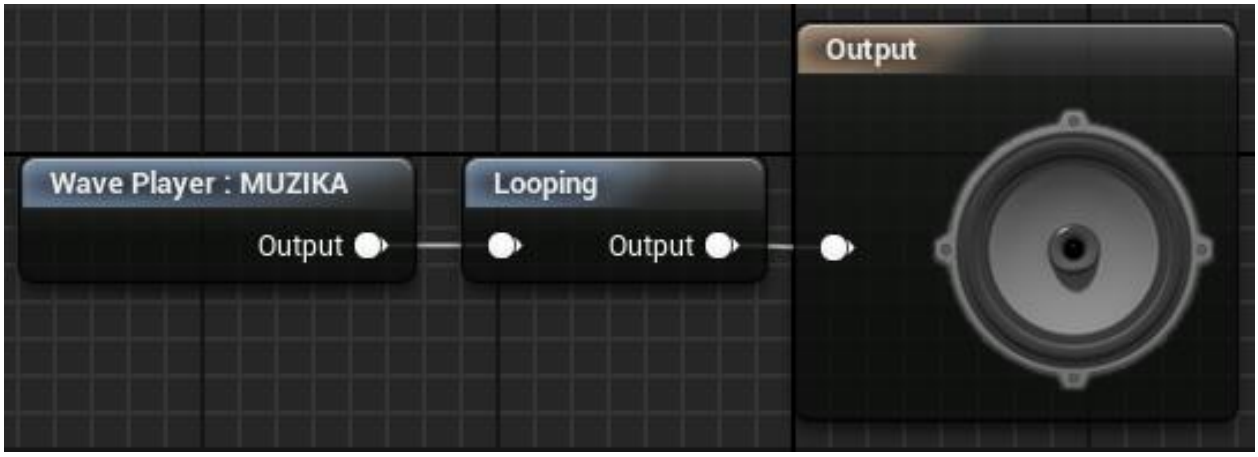
Kako bi se održala tema Pac-mana i mogli razlikovati neprijatelji svaki će imati materijal različite boje. Odabirom materijala BallMat koji je prije napravljen za kuglu desnim klikom se odabire opcija Duplicate i novi materijal naziva DuhMat1. Taj materijal se duplicira još 3 puta tako da ih se dobije 4. Ulaskom u svaki materijal mjenja se vrijednost Constant3Vectora koji je uključen u Base Color. Materijali se postavljaju na boje crvena , plava, narančasta i roza. Nakon što se izrade 4 različita materijala vraća se u blueprinte od neprijatelja i u Details tabu sa odabranom StaticMesh komponentom pod kategorijom Materials mjenja se Element 1 u materijal koji se prethodno izradio. Svakome se postavlja drugačiji. Element 0 je materijal koji je importan iz blendera gdje je bio crne boje za oči i to u ovom slučaju netreba mjenjati.



Slika 8.35 Pac-man

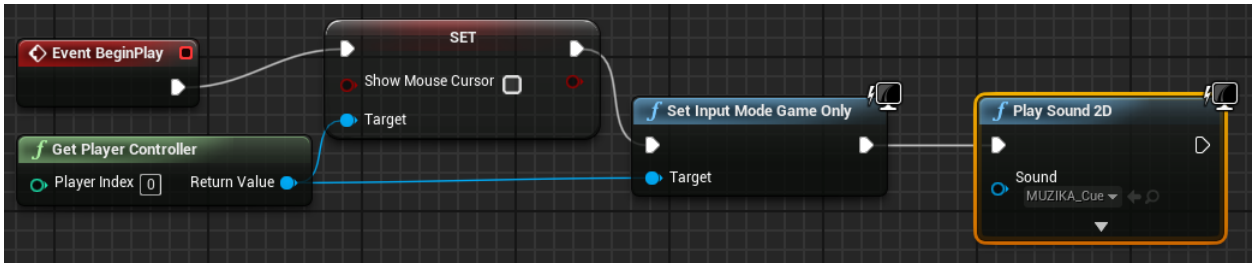
Zvučno oblikovanje je dodano na samom kraju igre. Igri se nisu dodavali zvučni efekti već samo pozadinska muzika koja se pokreće kada se učita level. Glazbena datoteka je pronađena na freesound.org pod nazivom Arcade Game Loop. Za pozadinsku glazbu najbolja opcija je odabrati izvedbu koja se „loopa“ što znači da se kraj i početak glazbene datoteke poklapaju i prilikom ponavljanja nebi smjelo biti primjetno gdje je kraj a gdje početak. Preporučljivo je da se pozadinska glazba loopa zato jer se nemože predvidjeti koliko će igrač vremena provesti u igri.

Za korištenje svi zvukovi moraju biti u .wav formatu za UE4. Zvukovi se uvoze tipkom Import. Nakon što se zvuk uveo desnim klikom na njega i odabirom Create Sound Cue stvara se zvučna komponenta koja se može koristiti u igri. Dvoklikom na tu novu komponentu otvara se editor zvuka koji isto radi kao i većina editora na principu blueprinta. U ovom slučaju između je ubačen samo element Looping kako bi se zvuk ponavljao.



Slika 8.36 Audio Editor

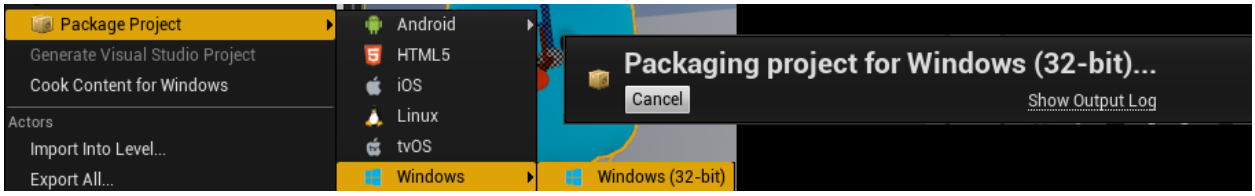
Nakon toga trebalo je odrediti da se glazba pokrene prilikom učitavanja levela. Logika se dodaje u level blueprintu na Event Begin Play. Od nekoliko različitih elemenata za reprodukciju zvuka odabran je Play Sound 2D jer se kod njega prilikom reprodukcije reproducirani zvuk jednako čuje na svim djelovima scene dok su drugi relativni na prostor u koji ga se postavi na sceni.



Slika 8.37 Play Sound 2D

Sada se igra smatra gotovom i može se krenuti sa pripremom igre za pokretanje na željenoj platformi.

Igra se zapakira odabirom File/Package Project/Windows 32-bit ili Windows 64-bit. U ovom slučaju odabran je Windows 32-bit zbog nepoznatih specifikacija računala na kojem će igra biti pokrenuta prilikom obrane ovog rada pa je sigurje zapakirati projekt za računalo slabijih specifikacija. Nakon toga bira se lokacija mape gdje će se igra pripremiti i čeka se podulje vrijeme da se to izvrši pri čemu se koristi mnogo računalnih resursa. Po završetku pakiranja zvučni signal javlja da je gotovo i igra se može locirati u odredišnoj mapi te pokrenuti i igrati.



Slika 8.38 Package Project

9. Zaključak

U radu je prikazano kako izgleda proces izrade videoigre od početka do kraja i objašnjeni su alati i ekosustav u kojima bi se svatko tko se nađe u ovom poslu najvjerojatnije susreo. Rad je rađen u nadi da se svakome tko je zainteresiran za ovo područje predoči kako funkcionira cjeli sustav i kako se videoigre izrađuju. Generalno nema previše odstupanja od ovog procesa jer se svi game enginei rade sa istim ciljem na umu, a to je da olakšaju i ubrzaju proces izrade.

Trenutno stanje u industriji vodi Steam na području platforme a Unity na području game enginea no zntni napredak vidljiv je u UE4 kao u tehnologiji tako i u zajednici koja konstantno raste i netko tko želi graditi karijeru u game developementu nebi pogriješio ako bi se počeo baviti s Unreal Engineom kojeme se predviđa još dugi životni vijek kao i Steamu jer kokurencije još nema na vidiku.

U Varaždinu,_____

_____

10. Literatura

Internet izvori:

[1] https://www.blender.org/about/, kolovoz, 2017.

[2] https://en.wikipedia.org/wiki/Blender_(software), kolovoz, 2017.

[3] https://rowvr.co/2015/01/03/get-from-blender-to-unreal-engine-4-quick-workflow-tips/, kolovoz, 2017.

[4] http://media.steampowered.com/apps/valve/Valve_NewEmployeeHandbook.pdf kolovoz, 2017.

[5] http://www.valvesoftware.com/company/index.html, kolovoz, 2017.

[6] https://www.epicgames.com/about,kolovoz, 2017.

[7] https://docs.unrealengine.com/latest/INT/Engine/Blueprints/, kolovoz, 2017.

[8] https://docs.unrealengine.com/latest/INT/Engine/Blueprints/Glossary/index.html, kolovoz, 2017.

[9] http://store.steampowered.com/news/8761/, kolovoz, 2017.

[10] https://steamcommunity.com/games/593110/announcements/detail/1328973169870947116, kolovoz, 2017.

[11] https://docs.unrealengine.com/latest/INT/Engine/UI/LevelEditor/, kolovoz, 2017.

[12] https://www.superdataresearch.com/market-data/market-brief-year-in-review/, kolovoz, 2017.

[13] http://www.valvesoftware.com/company/, kolovoz, 2017.

[14] https://www.gamasutra.com/view/news/213517/Epic_radically_changes_licensing_model_for_Unreal_Engine.php, kolovoz, 2017.

[15] https://www.unrealengine.com/previous-versions/udk-licensing-resources?prox=1#faq, kolovoz, 2017.

[16] https://thenextweb.com/gaming/2016/03/24/engine-dominating-gaming-industry-right-now/#.tnw_OJPsXTT9, kolovoz, 2017.

[17] https://www.unrealengine.com/en-US/blog/ue4-is-free, kolovoz, 2017.

[18] https://docs.unrealengine.com/latest/INT/Engine/Blueprints/TechnicalGuide/NativizingBlueprints/, kolovoz, 2017.

[19] http://counterstrike.wikia.com/wiki/Counter-Strike, kolovoz, 2017.

[20] https://stackoverflow.com/questions/27180450/visual-scripting-vs-coding, kolovoz, 2017.

[21] https://www.polygon.com/2017/8/8/16116786/valve-artifact-dota-2-card-game, kolovoz, 2017.

[22] https://www.quora.com/Valve-company-What-percentage-does-Steam-keep-from-sales, kolovoz, 2017.

[23] https://www.lifewire.com/top-pc-game-digital-download-services-813065, kolovoz, 2017.

[24] https://rowvr.co/2015/01/03/get-from-blender-to-unreal-engine-4-quick-workflow-tips/, kolovoz, 2017.

[25] https://en.wikipedia.org/wiki/Game_development_kit, kolovoz, 2017.

[26] https://medium.com/master-of-code-global/app-store-vs-google-play-stores-in-numbers-fd5ba020c195, kolovoz, 2017.

[27] https://steamcommunity.com/games/593110/announcements/detail/1265921510652460726, kolovoz, 2017.

[28] https://docs.blender.org/manual/en/dev/render/opengl.html, kolovoz, 2017.

11. Popis slika

Slika 5.1 UE4 FBX Import Options 15
Slika 7.1 Proces izrade videoigre 20
Slika 8.1 Unreal Project Browser 21
Slika 8.2 Unreal Editor 22
Slika 8.3 PM_Ball 23
Slika 8.4 PhysicsBallBP 23
Slika 8.5 Input Settings 24
Slika 8.6 Dodavanje elemenata 25
Slika 8.7 Kontrola kamere 25
Slika 8.8 Modifikacija logike iz predloška 26
Slika 8.9 Logika Neprijatelja1 27
Slika 8.10 Logika Neprijatelja2 28
Slika 8.11 Logika Neprijatelja3 28
Slika 8.12 Logika Neprijatelja4 30
Slika 8.13 Problematične rampe 30
Slika 8.14 Stožci 31
Slika 8.15 Bod oblikovanje 31
Slika 8.16 Logika Boda 31
Slika 8.17 Funkcija Dodaj Bod 32
Slika 8.18 Pozivanje funkcije Dodaj Bod 32
Slika 8.19 Widget Blueprint 33
Slika 8.20 Text Bodovi 33
Slika 8.21 Bind 34
Slika 8.22 Get Text 34
Slika 8.23 Add to Viewport 34
Slika 8.24 Prikaz bodova na sučelju 34
Slika 8.25 MeniLevel i MeniWidget 35
Slika 8.26 MeniWidget 35
Slika 8.27 OnClicked Event 35
Slika 8.28 MeniLevel Blueprint 36
Slika 8.29 Set Input Mode 36
Slika 8.30 Character Overlap 36

Slika 8.31 Bod Materijal 37
Slika 8.32 Modeliranje duha 38
Slika 8.33 Export FBX 38
Slika 8.34 Static Mesh 39
Slika 8.35 Pac-man 39
Slika 8.36 Audio Editor 40
Slika 8.37 PlaySound2D 40
Slika 8.38 Package Project 41

Sveučilište Sjever

SVEUČILIŠTE SJEVER

IZJAVA O AUTORSTVU
I
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Sandi Rahmani *(ime i prezime)* pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom Izrada 3D igre u Unreal Engine 4 i plasiranje na Steam platformu *(upisati naslov)* te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)

Sandi Rahmani
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnost i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, Sandi Rahmani *(ime i prezime)* neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (*obrisati nepotrebno*) rada pod naslovom Izrada 3D igre u Unreal Engine 4 i plasiranje na Steam platformu *(upisati naslov)* čiji sam autor/ica.

Student/ica:
(upisati ime i prezime)

Sandi Rahmani
(vlastoručni potpis)