

Proces razvoja videoigara

Petrović, Sven

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:211:079106>

Rights / Prava: [Attribution-ShareAlike 3.0 Unported](#)/[Imenovanje-Dijeli pod istim uvjetima 3.0](#)

Download date / Datum preuzimanja: **2025-02-12**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Sven Petrović

PROCES RAZVOJA VIDEOIGARA

DIPLOMSKI RAD

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ź D I N

Sven Petrović

Matični broj: 45983/17-R

Studij: Informacijsko i programsko inženjerstvo

PROCES RAZVOJA VIDEOIGARA

DIPLOMSKI RAD

Mentor:

Izv. prof. dr. sc. Mario Konecki

Varaždin, rujan 2024.

Sven Petrović

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Ovaj diplomski rad temelji se na procesu kreiranja avanturističke platformer 3D video igre za dva igrača uz pomoć nacrtu za razvoj unutar softverskog programa Unreal Engine 5. Visoko razvijeni alati i značajke softvera omogućuju razvojnim programerima brže i efikasnije plasiranje video igara na tržište. Svaka videoigra donosi nove i autentične izazove kako sa grafičkog aspekta tako i sa implementacijskog i produkcijskog.

Svrha ovog projekta je da se kroz praktični dio kreiranja videoigre suočimo s izazovima s kojima se jedan razvojni programer susreće, damo osvrt na sam proces i korake razvijanja video igre, njene priče te plasiranja na tržište.

Ključne riječi: Razvoj, proces, video igre

1 Sadržaj

1	Uvod	1
2	Metode i tehnike rada	2
3	Razrada teme	3
3.1	Izvor inspiracije za razvoj videoigre.....	3
3.2	Faze procesa razvoja videoigara.....	4
3.2.1	Učinkovito Planiranje.....	4
3.2.1.1	Brainstorming za izradu plana	4
3.2.1.2	Istraživanje tržišta.....	4
3.2.1.3	Izrada preliminarnog budžeta i rasporeda.....	5
3.2.1.4	Sastavljanje tima	6
3.2.2	Faza prije produkcije	6
3.2.2.1	Svrha faze	6
3.2.2.2	Dokumentacija i izrada prototipa	6
3.2.2.3	Definiranje tehničkih zahtjeva	7
3.2.2.4	Izrada detaljnog plana razvijanja proizvoda	8
3.2.2.5	Postavljanje alata i okruženja za razvoj	8
3.2.3	Produkcija u teoriji.....	8
3.2.3.1	Kodiranje mehanike videoigre, funkcionalnosti i značajki.....	8
3.2.3.2	Izrada 3D modela, animacija i resursa.....	8
3.2.3.3	Razvijanje svijetova i dizajna okruženja igrača	8
3.2.3.4	Integracija audio i vizualnih elemenata	9
3.2.3.5	Kontinuirana optimizacija.....	9
3.2.4	Produkcija u praksi.....	10
3.2.4.1	Kodiranje mehanike videoigre, funkcionalnosti i značajki.....	10
3.2.4.1.1	Mehanike kretanja glavnih likova	10
3.2.4.1.2	Mehanike odabira oružja i uporaba kolekcije likova	14
3.2.4.1.3	Razvijanje glavnih mehanika ponašanja likova.....	18
3.2.4.1.4	Kreiranje neprijateljskih funkcionalnosti	30
3.2.4.1.5	Animacijske video sekvence	32
3.2.4.1.6	Ostale funkcionalnosti i značajke.....	34
3.2.4.2	Odabir i izrada 3D modela i animacije	40
3.2.4.2.1	3D modeli likova	40
3.2.4.2.2	3D modeli oružja	41
3.2.4.2.3	3D modeli igrača i neprijatelja.....	42

3.2.4.2.4	Kreiranje vlastitog 3D modela	43
3.2.4.2.5	Preusmjeravanje animacije (<i>eng. Retargeter</i>)	46
3.2.4.3	Razvijanje svjetova i dizajna okruženja igrača	47
3.2.4.4	Integracija zvuka s vizualnim efektima.....	53
3.2.5	Testiranje	56
3.2.5.1	Osiguranje kvalitete (<i>eng. Quality Assurance Testing</i>).....	56
3.2.5.2	Otklanjanje pogrešaka (Debugiranje)	56
3.2.5.3	Testiranje performansi i optimizacija.....	57
3.2.5.4	Dokumentacija i izvještavanje.....	58
3.2.5.5	Beta i ostale verzije testiranja	60
3.2.6	Faza prije lansiranja	61
3.2.6.1	Marketing	61
3.2.6.2	Postavljanje sustava podrške	61
3.2.7	Lansiranje	61
3.2.7.1	Distribucija videoigre, praćenje i rješavanje problema	61
3.2.8	Faza nakon lansiranja	62
3.2.8.1	Praćenje povratnih informacija igrača i ispravak grešaka nakon lansiranja ..	62
3.2.8.2	Planiranje, razvoj i nadogradnja dodatnog sadržaja	62
3.2.9	Lansiranje videoigre na tržište	63
3.2.9.1	Platforma Steam.....	63
3.2.9.2	Pravila i koraci lansiranja.....	63
3.2.9.3	Mrežni podsustav STEAM-a i API.....	64
4.	Zaključak	66
5.	Popis literature.....	67
6.	Popis slika	70
7.	Popis tablica	73

1 Uvod

Videoigra je specifična vrsta softvera implementirana unutar operacijskog sustava u kojoj korisnici komuniciraju s korisničkim sučeljem i ostalim ulaznim uređajima kako bi ostvarili jedinstvene ciljeve i doživljaje unutar virtualnog okruženja koje im se povratno projektira u vizualnom formatu na ekranu [1]. Videoigre obuhvaćaju razne žanrove i stupnjeve složenosti, od jednostavnih arkadnih igara do kompleksnih simulacija i narativnih avantura s kojom se u ovom radu i bavimo.

Svrha videoigara je pružiti zabavu i interaktivno iskustvo igračima kroz virtualne svjetove. One služe kao sredstvo za bijeg od svakodnevnog života, omogućujući igračima da se opuste, natječu ili kreativno izraze. Videoigre također mogu poticati kognitivni razvoj, vještine rješavanja problema i suradnju među igračima. Uz to, često se koriste u obrazovne svrhe i terapijske procese, pružajući inovativne metode učenja i liječenja [2]. Također, videoigre omogućuju društvenu interakciju putem online zajednica i „multiplayer“ načina igre.

Proces razvoja videoigara sastoji se od nekoliko ključnih faza koje osiguravaju stvaranje kvalitetnog i zabavnog proizvoda. U ovom radu dotaknut ćemo se ključnih faza, pojasniti njihovu ulogu i ciljeve zajedno kroz praktični rad i vlastito iskustvo koje sam stekao stvarajući ovaj projekt. Kreiranjem vlastite igre razvojni programeri moraju proći kroz faze konceptualizacije, preprodukcije i produkcije te fazu testiranja i lansiranja proizvoda na tržište (uz dodatnu podršku prilikom lansiranja). Unutar rada najviše pažnje posvetit ćemo glavnoj fazi, a to je produkcija. Praktični dio projekta realiziran je uz pomoć softvera za razvoj video igara i animacija Unreal Engine 5. Jedan od razloga zašto sam se odlučio koristiti ovim softverom pri izradi projekta je to što nova verzija Unreal Engine-a donosi nekoliko naprednih nikad prije viđenih značajki, uključujući „Nanite“ (sustav za virtualnu geometriju i modele), Lumen (osvjetljenje), Mass AI za simulaciju velikog broja AI likova istovremeno sl [3].

Kao jedan od korisnika ovakvog oblika zabave, motiviralo me da i sam izradim videoigru i iznesem svoje iskustvo stečeno tijekom procesa razvijanja ovog projekta. Stoga, izuzev opisivanja i obrade teorijskog dijela samog procesa razvoja videoigara cilj ovog diplomskog rada je također na vlastitom primjeru potvrditi i osjetiti na vlastitoj koži neke od ciklusa izrade videoigre te iz njega izvući vlastiti zaključak.

2 Metode i tehnike rada

Kao što je ranije navedeno za izradu praktičnog dijela ovog diplomskog rada koristio se programski alat Unreal Engine 5.

Unreal Engine 5 nudi dva glavna načina za programiranje a to su nacrti (*eng. Blueprints*) i C++ programski jezik. Nacrti su vizualni skriptni sustav koji omogućava korisnicima bez programerskog iskustva stvaranje logike igre pomoću čvorova, dok je C++ moćan jezik koji omogućava dublju kontrolu nad performansama i složenijim sustavima. Nacrti su idealni za brzu iteraciju i prototipove, ali mogu biti manje učinkoviti u složenijim projektima. C++ nudi veću fleksibilnost i optimizaciju, ali zahtijeva više vremena i stručnosti. Često se koristi kombinacija, gdje nacrti služe za brze promjene i dizajnerske potrebe, dok C++ upravlja ključnim funkcionalnostima [4].

Često se koristi kombinacija oba pristupa: nacrti za brze promjene i dizajnerske potrebe, dok C++ upravlja ključnim funkcionalnostima. Da bi C++ klase i funkcije bile dostupne u Nacrtima, potrebno ih je označiti s „UFUNCTION()“ i „UPROPERTY()“. Ova integracija omogućuje brzu prilagodbu i testiranje kroz nacрте, dok C++ osigurava optimizaciju performansi i složenije algoritme. Ovaj pristup omogućuje učinkovitu suradnju između programera i dizajnera, kao i modularan razvoj igre, čime se postiže bolja fleksibilnost i brzina razvoja [4].

Zbog praktičnosti i svoje prvenstveno jednostavnosti i dostupnosti ovaj diplomski rad temeljio se na nacrtima. Dodatni programski alat koji je sudjelovao u izradi praktičnog djela je Blender. To je softver za 3D modeliranje, animaciju, renderiranje i video montažu. Prvenstveno se koristio za 3D modeliranje i prijenos modela iz jednog formata u drugi kako bi bio podržan od strane Unreal Engine-a [5].

Prilikom izrade videoigre od velike pomoći također su bili izvori informacija prikupljeni na raznim videozapisima na „Youtube“ kanalima te tečaj dostupan na Udemy web stranici [6].

3 Razrada teme

Prilikom razvijanja videoigre ili bilo kojeg proizvoda, primjerice u industriji, možemo im pronaći nešto zajedničko, a to je sam proces razvoja i stvaranja. Razvoj video igara je proces stvaranja koji rezultira novom videoigrom na tržištu. Riječ je o složenom proizvodnom procesu u kojem kreativni tim dizajnera i programera prolazi kroz sve faze razvoja video igre kako bi pretvorio ideju o novoj igri u gotov komercijalni proizvod.

Ovaj proces kako bi bio dobro organiziran, svaki član tima zna što treba raditi i kada. Scenaristi, umjetnici, dizajneri, programeri, producenti, inženjeri i menadžeri svjesni su svih faza razvoja igre kako bi razumjeli ulogu svakog pojedinca u procesu.

Za razliku od programera i dizajnera videoigara, mnogi kupci/igrači nemaju pravu sliku o tome kako nastaje nova videoigra. Vrijeme razvitka i budžet obično su izvan njihovog fokusa. Svaki potpuno razvijen tim za razvoj videoigara sastoji se od različitih stručnjaka s posebnim ulogama koje su ključne za svaku fazu razvoja igre. Razmjer projekta određuje veličinu i sastav tima. Osim toga, bilo koji osnovni tim može uključivati različite studije koji pružaju dodatne usluge kako bi se završio određeni dio razvojnih cjelina [7].

U nastavku slijedi dublja obrada svake od razvojnih faza, izvor motivacije i kako određeni alati i internet danas pomažu pojedincima u stvaranju proizvoda i rezultata.

3.1 Izvor inspiracije za razvoj videoigre

Inspiracija za razvoj i dizajniranje videoigre prvobitno je bila ljubav prema videoigramima od malena a zatim i žanrovi igara poput akcijskih i avanturističkih gdje su neke od njih namijenjene i za više igrača:

- Assassin's Creed
- Prince of Persia
- It Takes Two
- Final Fantasy
- Grand Theft Auto

Ono što ove videoigre imaju zajedničko s praktičnim projektom diplomskog rada su elementi razrade priče i animacije koje daju određeni ambijent, akcijske borbe s mogućnošću odabira oružja i lasera za eliminaciju neprijatelja, platformerske mape koje zahtijevaju određenu vještinu upravljanja kontrolama te je namijenjena za 2 igrača.

3.2 Faze procesa razvoja videoigara

Produkcija igara je dugotrajan proces koji se često uspoređuje s cjevovodom (eng. pipeline) jer se proizvodnja mora odvijati u specifičnom redoslijedu kako bi bila učinkovita i uspješna na kraju procesa. Opseg cjevovoda za proizvodnju igara može se znatno razlikovati ovisno o veličini produkcijskog i razvojnog tima te platformi na kojoj će igra biti objavljena.

Međutim, općenite faze razvoja videoigara podijeljene su u tri glavna dijela: faza prije produkcije, produkcija i faza nakon produkcije, koje ćemo detaljno razraditi zajedno s ostalim fazama koje nisu ništa manje bitne od glavnih faza kako bismo dodatno definirali cjevovod [8].

Kod određenih faza dotaknut ćemo se i praktičnog dijela rada kako bi upotpunili neke od teza i potvrdili koliko je ova faza važna za daljnji razvoj.

3.2.1 Učinkovito Planiranje

3.2.1.1 *Brainstorming za izradu plana*

Bez obzira na vrstu igre, planiranje obično započinje (eng. brainstorming) sesijama u kojima definiramo kako najučinkovitije i najisplativije implementirati zahtjeve klijenta. Također analiziramo tržišne trendove, preferencije igrača te razmatramo različite platforme. Cilj je izraditi sveobuhvatan plan za isporuku naprednog rješenja koje je usklađeno s vizijom i ciljevima klijenta [9].

Prilikom istraživanja, kako navodi globalna razvojna tvrtka Pingle, uspostavilo se da je u 2023. godini najtraženija platforma bila stolno računalo. Naime računala postaju nešto pristupačnija otkako su cijene grafičkih kartica pale zbog smanjenog interesa za rudarenje kriptovaluta. Iako je nestašica poluvodiča još uvijek prilično ozbiljna, cijene nisu toliko visoke kao prije nekoliko godina [10].

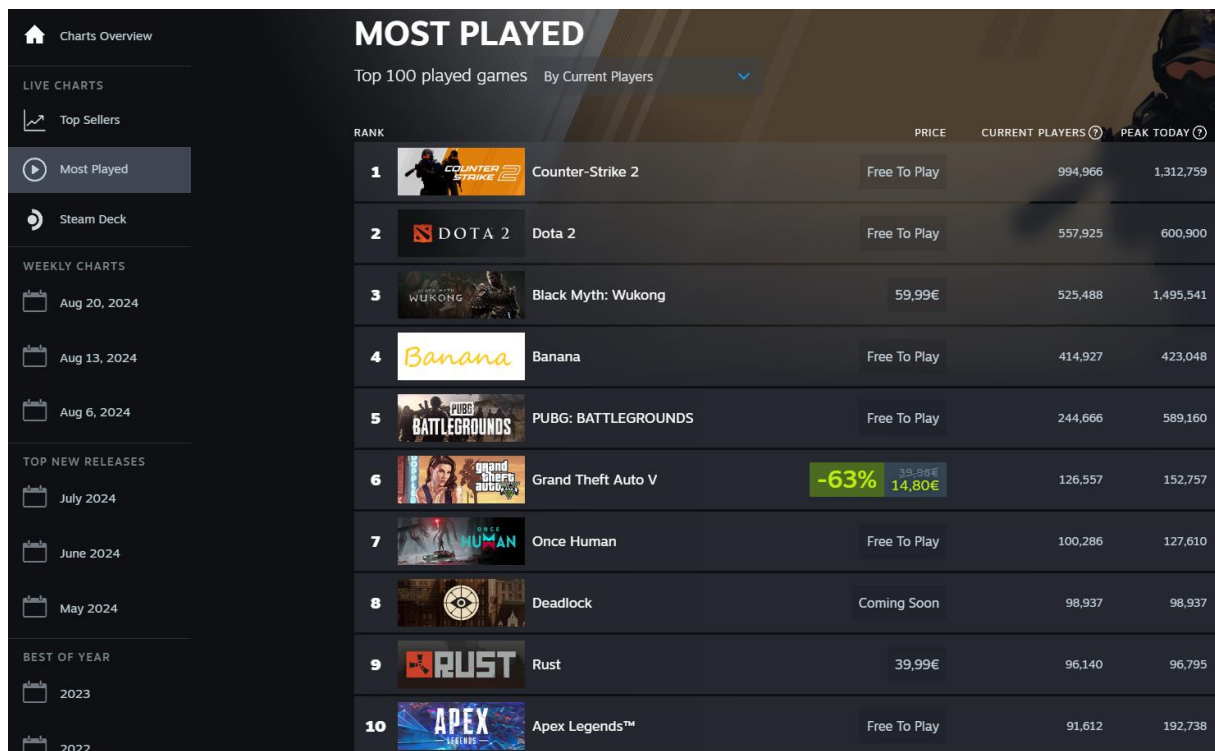
Stoga, iz gore navedenoga zaključujemo kako bi najbolje bilo razvijati videoigru koja prvenstveno podržava PC i laptop a zatim ostale platforme što ćemo vidjeti kasnije da je i realizirano.

3.2.1.2 *Istraživanje tržišta*

U ovoj fazi detaljno se analiziraju opći trendovi unutar industrije i ponude konkurencije. Pokušavaju se identificirati moguće praznine na tržištu i planira se kako iskoristiti te prilike. Rezultat ove podfaze je jasnije razumijevanje tržišnog okruženja i strategije za razvoj igre koja će se istaknuti na tržištu.

Primjerice, u ovome projektu, ono što veliki broj konkurencije danas nudi je videoigra za više igrača (eng. *Multiplayer*) koje su se pokazale vrlo uspješne unazad

nekoliko godina te dokaz pronalazimo na jednoj od najpoznatijih „gaming“ platformi za videoigre:



RANK	PRICE	CURRENT PLAYERS	PEAK TODAY
1	Free To Play	994,966	1,312,759
2	Free To Play	557,925	600,900
3	59,99€	525,488	1,495,541
4	Free To Play	414,927	423,048
5	Free To Play	244,666	589,160
6	39,99€ -63% 14,80€	126,557	152,757
7	Free To Play	100,286	127,610
8	Coming Soon	98,937	98,937
9	39,99€	96,140	96,795
10	Free To Play	91,612	192,738

Slika 1 Najigranije videoigre na platformi STEAM (Izvor: <https://store.steampowered.com/charts/mostplayed>)

Stoga je vrlo logično razviti takav tip igre i pronaći unikatnu priču, likove, animacije koju bi određeni skup igrača zavolio i održavao na tržištu.

3.2.1.3 Izrada preliminarnog budžeta i rasporeda

Ova pod faza podrazumijeva procjenu financijskih sredstava potrebnih za razvoj i drugih povezanih troškova. Raspored se definira s određenim vremenskim rokovima za različite faze razvoja, pri čemu će svaka faza imati svoj set rokova i ciljeva.

Kako na ovom projektu radi pojedinac, planirani budžet i resursi koji se planiraju potrošiti su svedeni na minimum. Razlog tome je jasan, ali zato postoji puno dostupnog materijala koji se mogu preuzeti besplatno kao i razvojni alati.

3.2.1.4 *Sastavljanje tima*

Nakon što su sve prethodne faze i koraci dovršeni, potrebno je sastaviti pravi razvojni tim za proces razvoja video igre. Tim treba uključivati dizajnere, 2D i 3D umjetnike, VFX umjetnike, programere, zvučne inženjere i druge stručnjake.

Kako je tehnologija u zadnjih nekoliko godina napredovala, današnjim razvojnim programerima nudi se veliki broj alternativnih rješenja. Tako su se za ovaj projekt koristili gotovi tzv. proizvodi, odnosno likovi, animacije, SFX zvukovi, 3D modeli i ostali 2D vizualni dodaci koji stvaraju dodatni ambijent i čar videoigri koji su dostupni svima i smiju se koristiti u bilo koje svrhe.

3.2.2 **Faza prije produkcije**

3.2.2.1 *Svrha faze*

Cilj dizajnera je steći bolji uvid u ono što žele stvoriti prije nego što započnu samu izradu. Pred produkcija se može opisati kao kreativni prostor između koncepta i stvarne proizvodnje. Zato treba definirati "što, gdje i zašto" unutar videoigre. Važno je ne ulaziti previše u detalje u fazi pre-produkcije. Različiti timovi pristupaju problemu iz svojih specifičnih područja, nastojeći se uskladiti, što zahtijeva stalnu komunikaciju između disciplina [15].

3.2.2.2 *Dokumentacija i izrada prototipa*

Prototipiranje omogućava timu više prilika da testira predloženu mehaniku igre ili ideju te rano razumije igračev doživljaj igre. Imamo dvije vrste prototipa, eksplorativni i eksperimentalni.

Eksplorativni prototipovi koriste se za istraživanje novih ideja, alternativa ili identificiranje zahtjeva za sve što nije dobro definirano. Eksperimentalni prototipovi služe za validaciju sistemskih zahtjeva i pomažu dizajneru da prepozna snagu i slabosti u konceptu [11]. Testiranjem ovakvih prototipa sam proces se dokumentira kako bi se izbjegli unaprijed planirani problemi prilikom produkcije i lansiranja proizvoda.

3.2.2.3 Definiranje tehničkih zahtjeva

Uključuje određivanje hardverskih i softverskih specifikacija potrebnih za razvoj videoigre. U ovoj fazi donose se ključne odluke povezane sa specifikacijama računala, programskim jezicima i drugim tehničkim aspektima.

Primjerice, za ovaj projekt ključno je bilo predvidjeti mogu li se određeni programski alati pokretati i koristiti prilikom razvijanja videoigre. Tako su u ovom slučaju bili upotpunjeni ovakvi zahtjevi koje možemo vidjeti u idućoj tablici:

Tablica 1 Minimalni specifikacijski zahtjevi programskih alata (Izvori: [13] i [14])

Zahtjevi	Minimalni zahtjevi: Unreal Engine 5	Minimalni zahtjevi: Blender	Lenovo IdeaPad L340 Gaming
Operacijski sustav	Windows 10 ili više	Windows 10 ili više	Windows 10
Grafička kartica	DirectX 11 ili 12 kompatibilna kartica	2GB RAM	NVIDIA GeForce GTX 1650
Procesor	Quad-core Intel ili AMD, 2.5 GHz ili brži	Quad-core Intel ili brži	Intel (R) Core i5- 9300H CPU - 2.5 GHz
Radna memorija	8 GB RAM	8 GB RAM	8 GB RAM
Pohrana (tvrdi disk)	35-40 GB	500 MB	1 TB
Visual Studio	Visual Studio 2022	-	Visual Studio 2022

3.2.2.4 Izrada detaljnog plana razvijanja proizvoda

Plan razvijanja proizvoda je plan koji razlaže razvojni proces na niz manjih zadataka, služi kao vodič za tim, pomažući u organizaciji i postavljanju prioriteta posla. Uključuje vremenske rokove za različite zadatke, alokaciju resursa te ovisnosti između različitih komponenti igre [9].

3.2.2.5 Postavljanje alata i okruženja za razvoj

Na samom kraju, ova faza uključuje i odabir potrebitih alata i okruženja za razvoj. Ovaj korak obuhvaća konfiguraciju softvera, sustava za kontrolu verzija te platformi za suradnju koje će tim koristiti tijekom razvoja. Cilj je osigurati da je razvojno okruženje optimizirano za učinkovitost kako bi tijekom rada faze produkcije bio što fluidniji. Današnje razvojne kuće, gotovo svi teže već testiranim i popularnim softverskim pogonima za razvoj igara (eng. Game Engine) [12].

3.2.3 Produkcija u teoriji

3.2.3.1 Kodiranje mehanike videoigre, funkcionalnosti i značajki

U fazi produkcije, srce procesa razvijanja videoigre leži u kodiranju mehanike igre i značajki navedenih u GDD-u (eng. Game Design Document). Programeri započinju implementaciju sustava koji će definirati kako igra funkcionira [11]. Baza koda kontinuirano se usavršava dok razvojni tim dorađuje i optimizira iskustvo igranja kako bi ono u potpunosti odgovaralo zahtjevima klijenta.

3.2.3.2 Izrada 3D modela, animacija i resursa

Umjetnici i animatori rade na izradi 3D modela, animacija, tekstura i drugih vizualnih resursa koji su u skladu s konceptualnom umjetnošću razvijenom tijekom preprodukcije. Stalna komunikacija između umjetničkog i programerskog tima ključna je za osiguranje da vizualni elementi besprijekorno integriraju s mehanikom igre.

3.2.3.3 Razvijanje svjetova i dizajna okruženja igrača

Dizajneri mapa fokusiraju se na stvaranju različitosti u aspektu okruženja koje će igrači doživjeti. To uključuje oblikovanje rasporeda, postavljanje resursa i dizajniranje izazova koji će angažirati igrače na različite načine. Iterativno testiranje odvija se kako se razine razvijaju te se tako osigurava da pružaju željenu težinu, angažman i napredovanje igrača.

3.2.3.4 Integracija audio i vizualnih elemenata

Skladatelji i animatori zvuka integriraju audio elemente, poput glazbe, zvučnih efekata i glasovnih snimki. Ovo poboljšava uganjajuću kvalitetu igre i doprinosi cjelokupnom iskustvu igrača. Integracija audio i vizualnih elemenata je iterativni proces, s kontinuiranim usavršavanjem na temelju testiranja i povratnih informacija. Animatori za vizualne elemente igraju ključnu ulogu u stvaranju krajolika, ambijenata i atmosfere koja obogaćuje doživljaj igrača.

3.2.3.5 Kontinuirana optimizacija

Optimizacija je kontinuirani proces tijekom razvijanja proizvoda. Potrebno je stalno poboljšanje performansi, smanjivanje vrijeme učitavanja i osiguravanje da igra radi fluidno na različitim platformama. Također, tim može provoditi testiranje igranja i prikupljati povratne informacije igrača putem foruma kako bi identificirao područja koja se mogu dodatno optimizirati za poboljšanje kvalitete korisničke interakcije. Tehnike optimizacije mogu se koristiti s objektno orijentiranim programiranjem kako bi se izbjegli nepotrebni suvišni razredi i nasljeđivanje.

3.2.4 Produkcija u praksi

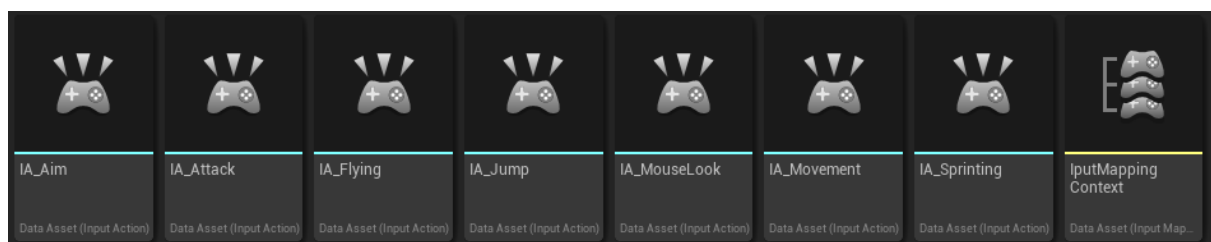
3.2.4.1 Kodiranje mehanike videoigre, funkcionalnosti i značajki

U ovom ulomku dotaknut ćemo se glavnih mehanika i funkcionalnosti videoigre koje su implementirane u svrhu praktičnog dijela diplomskog rada. To su:

- 1) Mehanike kretanja glavnih likova
- 2) Mehanike odabira oružja i uporaba kolekcije likova
- 3) Razvijanje glavnih mehanika ponašanja likova
- 4) Kreiranje neprijateljskih funkcionalnosti i ponašanje umjetne inteligencije
- 5) Animacijske video sekvence
- 6) Ostale funkcionalnosti i značajke

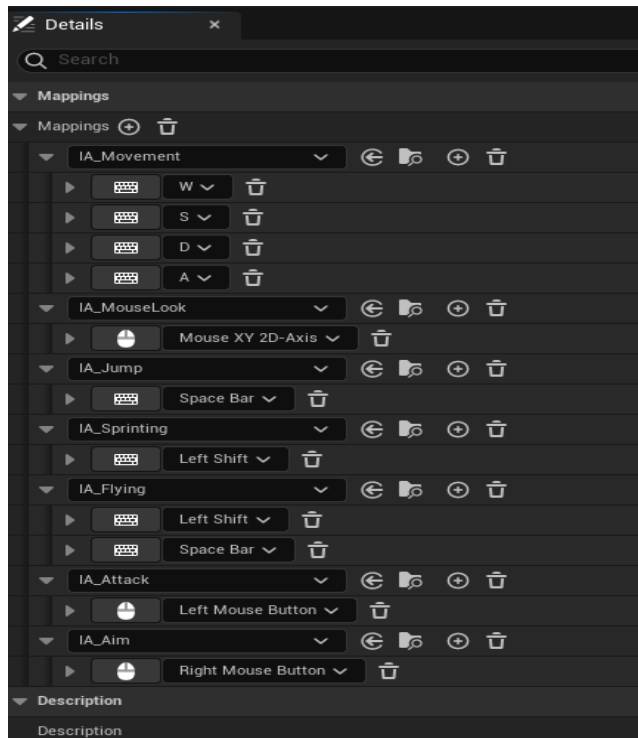
3.2.4.1.1 Mehanike kretanja glavnih likova

Cilj projekta bio je implementirati videoigru za 2 igrača čiji likovi bi imali „Third person“ perspektivu kamere. Programski alat Unreal Engine kao i u stariji verzijama tako i sada u najnovijoj nudi već unaprijed definirane postavke kontrola za kretanje likova. U same kontrole ubrajamo hodanje, trčanje, čučanj, skok, klizanje po podlozi, ležanje i sl. Prvi korak za realiziranje ovih kretnji u svijetu videoigre je definirati pravilne inpute, odnosno postaviti gumb za svaku akciju i interakciju s likom.



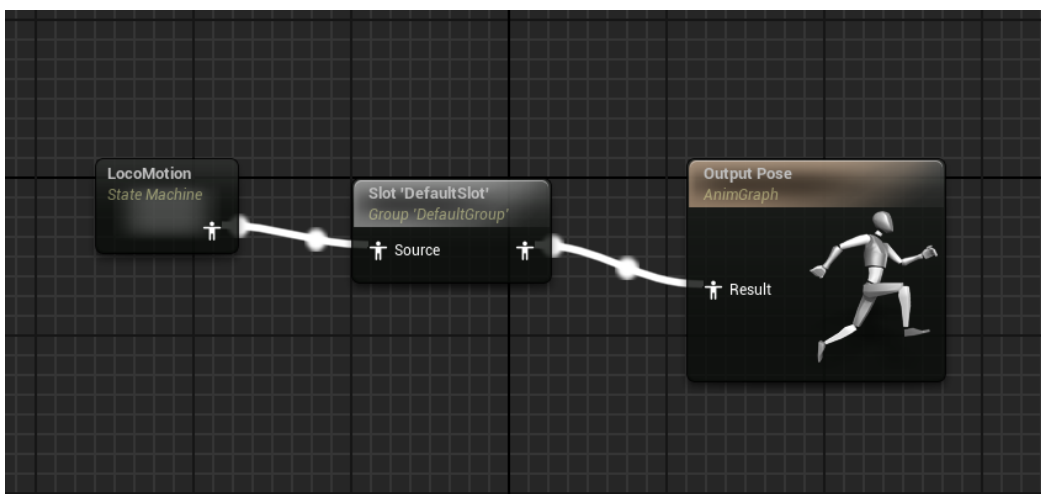
Slika 2 Input Action

Kada smo kreirali instancu željenih inputa, unutar kreiranog konteksta za mapiranje inputa definiramo gumb.



Slika 3 Mapiranje inputa

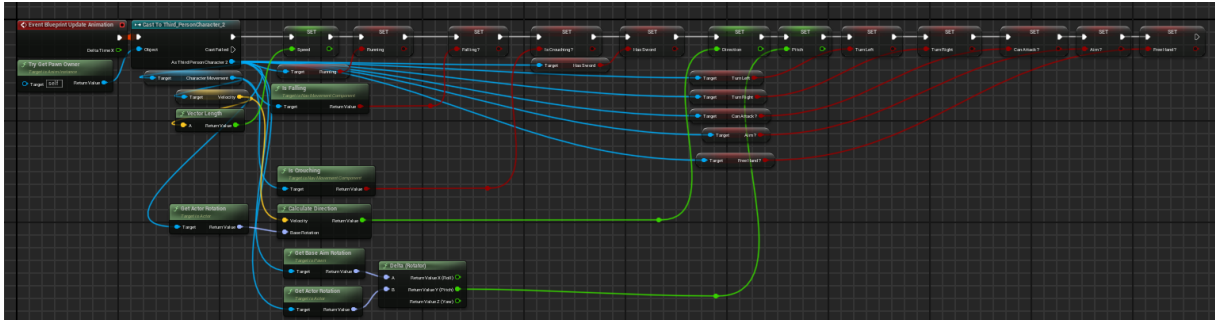
Sistem koji upravlja animacijama likova, posebno njihovim kretanjem naziva se „Locomotion“. On se koristi animacijskim nacrtima i stanjima mašine kako bi efikasno i fluidno prelazio iz jedne animacije u drugu. Funkcionalnosti ovih kretnji definirali smo u zasebnim klasama, odnosno nacrtima pod nazivom „ThirdPersonCharacter“ i „ThirdPersonCharacter2“.



Slika 4 Locomotion igrača 1

Zbog specifične animacije (pucanje lasera) možemo primijetiti kako se „Locomotion“ igrača 2 razlikuje od „Locomotion“-a igrača 1.

Kako bi se animacije skladno ažurirale, koristimo se nacrtima pod nazivom „Event Blueprint Update Animation“. Ovaj događaj analizira svaki „frame“ i koristi se za pravilno sinkroniziranje animacija s trenutnim stanjem lika. Ako se stanje lika promijeni tokom igre, te promjene mogu biti ažurirane u animaciji.



Slika 8 Nacrt - Ažuriranje animacija igrača 2

Vrlo važna stavka koju nam pruža Unreal Engine je ta što uz pomoć postavke „Blend Space“, animacije su postavljene na specifične točke na grafu ispod animacije lika na temelju kojih ova postavka zatim miješa animacije ovisno o X i Y vrijednostima (X – brzina lika, Y – smjer kretanja lika).

Z



Slika 9 "Blend Space" igrača 1

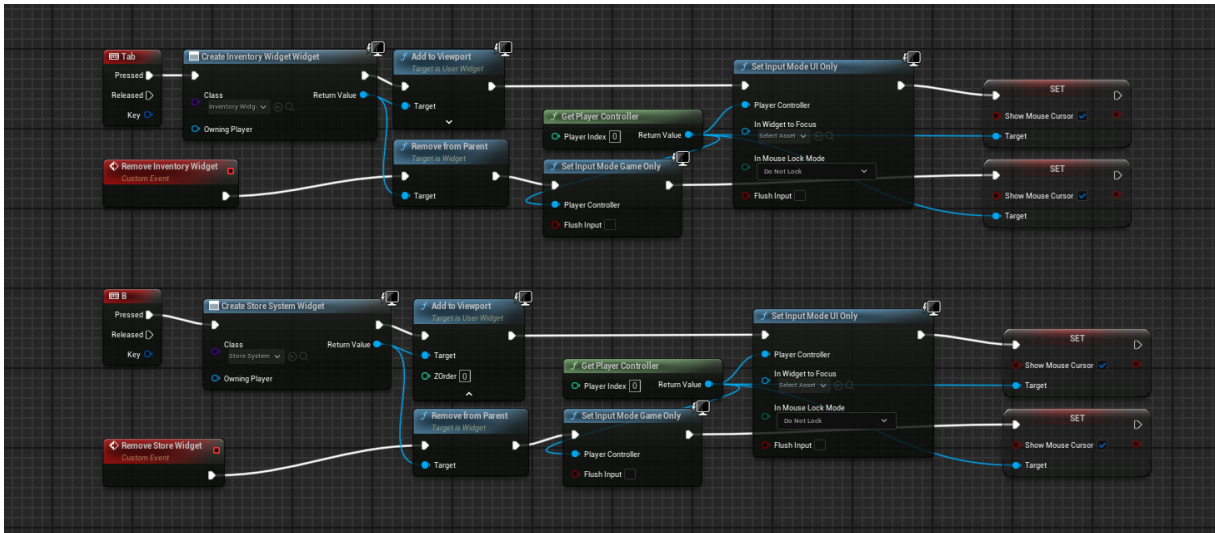
3.2.4.1.2 Mehanike odabira oružja i uporaba kolekcije likova

U početku, u videoigri moguć je odabir samo šaka kao izvora udarca i stvaranja štete. Kod igrača 1 (dječak) i igrača 2 (djevojčica) animacije udaraca razlikuju se te igrač 1 ovdje ima nešto jednostavniju mehaniku pripreme udarca dok igrač 2 prije udarca pritiskom desnim klikom miša podiže ruke, vrši animaciju a zatim udara.



Slika 10 Boksачki stav animacije igrača 2

Kasnije igrači dobivaju super napade i mogućnost korištenja mačeva i štita. Po standardu dodijeljena im je jedna vrsta mača no kupovinom u trgovini, novčićima mogu kupiti i ostale vrste mačeva. Mehanika i snaga za cilj ovog diplomskog rada kod svih mačeva funkcioniра isto. Igrač 1 ima mogućnost uporabe svoje kolekcije a nacrt i dizajn prikazani su u nastavku.



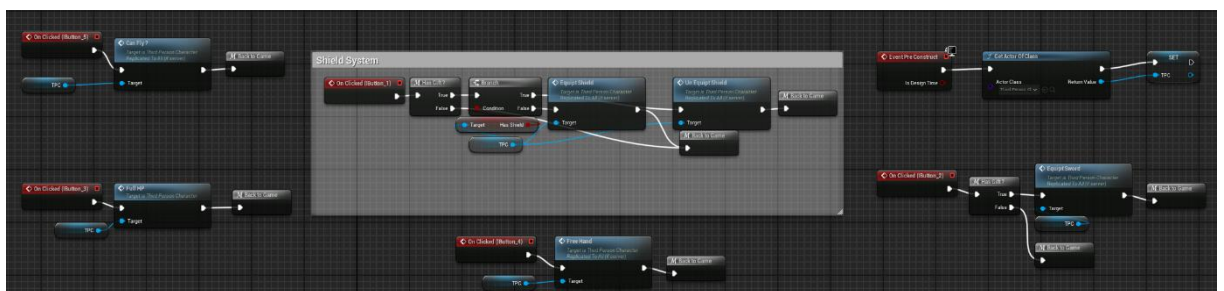
Slika 11 Nacrt pokretanja kolekcije i trgovine

Unutar kolekcije igrač može birati štit, mač, ozdravljenje, šake i mini propeler za letenje.



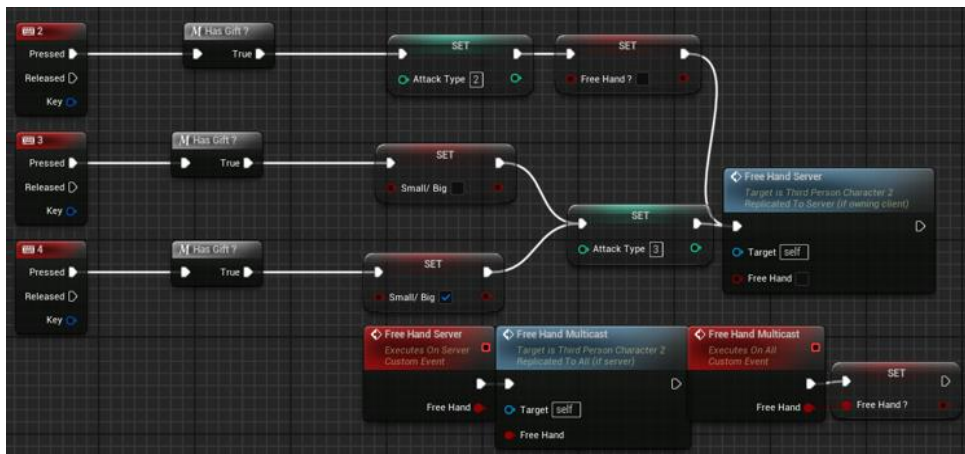
Slika 12 Kolekcija igrača 1

Ovisno o pritisku gumba dogodit će se određena aktivnost ili postavljanja oružja u ruke lika ili obrnuto, kao i postavljanje propelera na glavu.

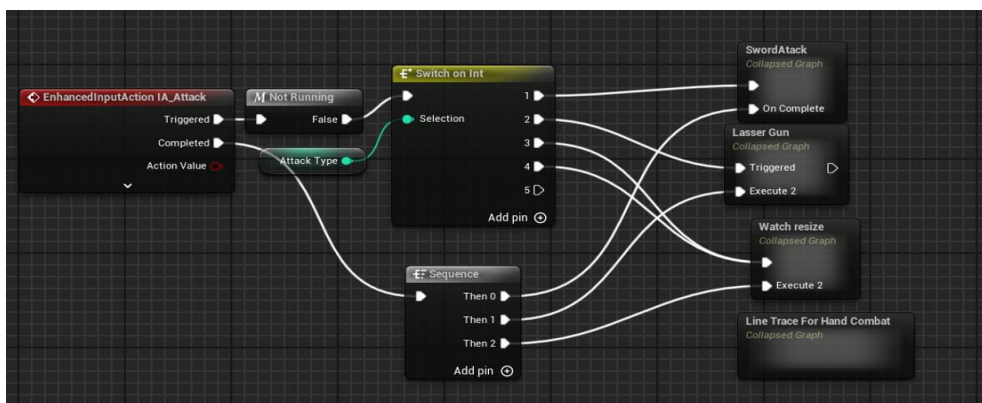


Slika 13 Nacrt kolekcija igrača 1

Mijenjanje oružja i objekata provodi se pritiskom određenih tipki te je implementirana mogućnost prikaza promjene oružja za glavni server (igrača 1) te za klijenta (igrača 2).

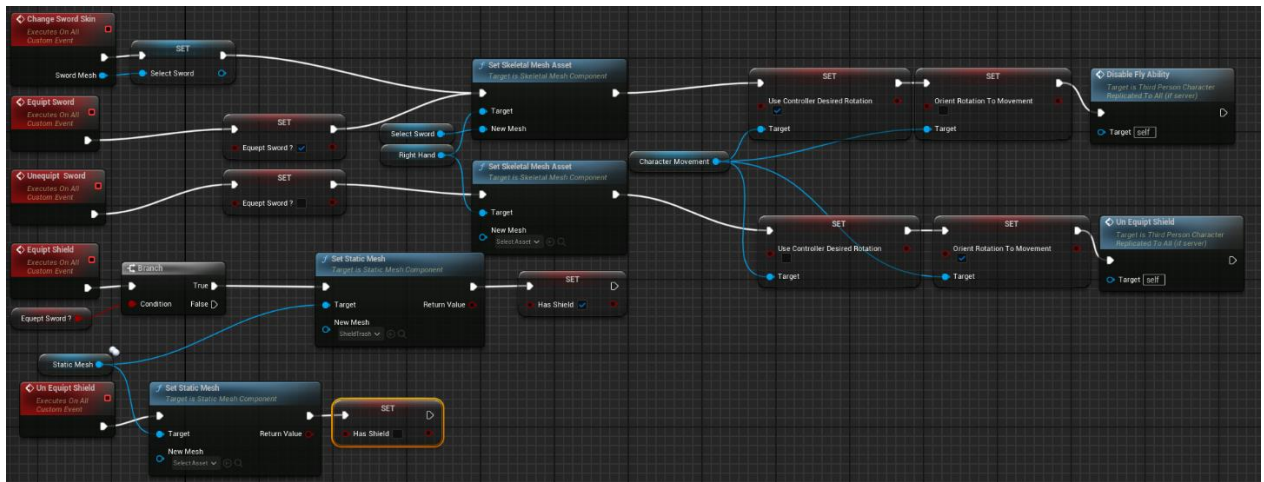


Slika 14 Nacrt promjene oružja i prikaz na serveru i klijentu (igrač 1)



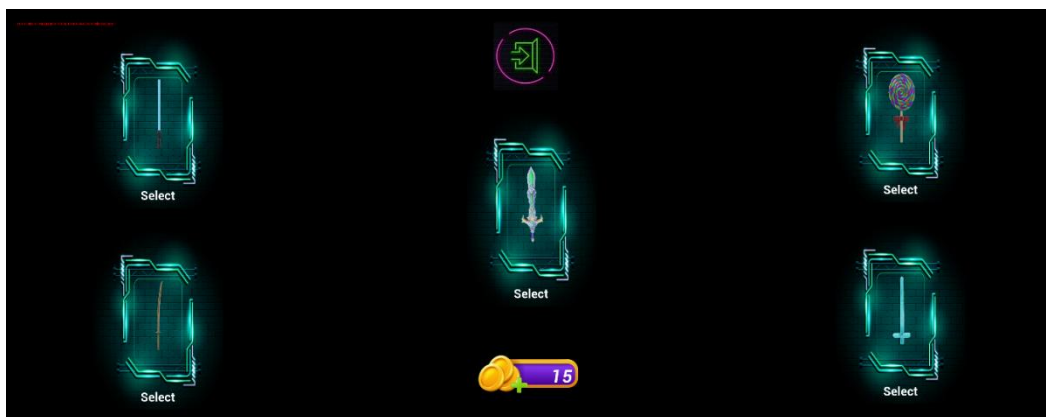
Slika 15 Nacrt inputa za promjenu oružja kod igrača 2

Prilikom implementacije nužno je voditi računa koje ostale mehanike bi likovi smjeli moći izvoditi, tako je u ovom slučaju isključena mogućnost letenja prilikom opremanja oružjem.



Slika 16 Nacrt implementacije opremanja oružjem s ograničenjem

Trgovina služi za kupnju raznih vrsta mačeva koje je moguće kupiti skupljanjem novčića na sporednim misijama i ulici.



Slika 17 Trgovina za igrača

Kada se pritisne gumb (broj 4), u ovom slučaju lizalica, tada se postavljaju parametri da igrač posjeduje ovu vrstu mača. Također se testiralo i pazilo na to da ako igrač nema dovoljno novčića ne može kupiti određeno oružje.



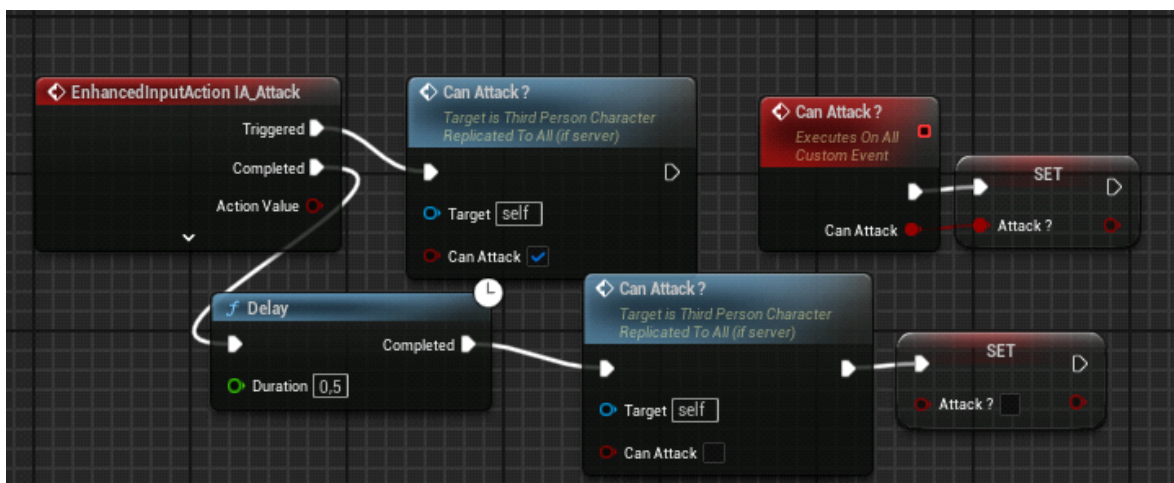
Slika 18 Nacrt za kupovinu oružja

3.2.4.1.3 Razvijanje glavnih mehanika ponašanja likova

Ovo poglavlje dotaknut će se glavnih mehanika s kojima se igrači susreću prilikom igranja. Pojedini lik ima svoje posebne mehanike te time oba igrača dobivaju različito iskustvo igranja. Glavne klase "Animation Blueprint" za likove naziva "Third person character" i "Third person character 2" ključne su za implementaciju svih potrebnih mehanika lika. Igrač 1, za dobrobit implementacije videoigre za dva igrača predstavlja server a igrač 2 gosta sesije.

3.2.4.1.3.1 Napad šakama

Kod dječakovog lika, animacije su jednostavnije implementirane nego kod lika djevojčice koja prvo podiže ruke u boksački stav a zatim napada nakon što smo pritisnuli potrebiti input. Kako bi realizirali ovu animaciju do kraja sa svim svojim postavkama, primjerice nakon lijevog klika miša pozivamo funkciju pod nazivom "EnhancedInputAction IA_Aattack".



Slika 19 Input napada šakom

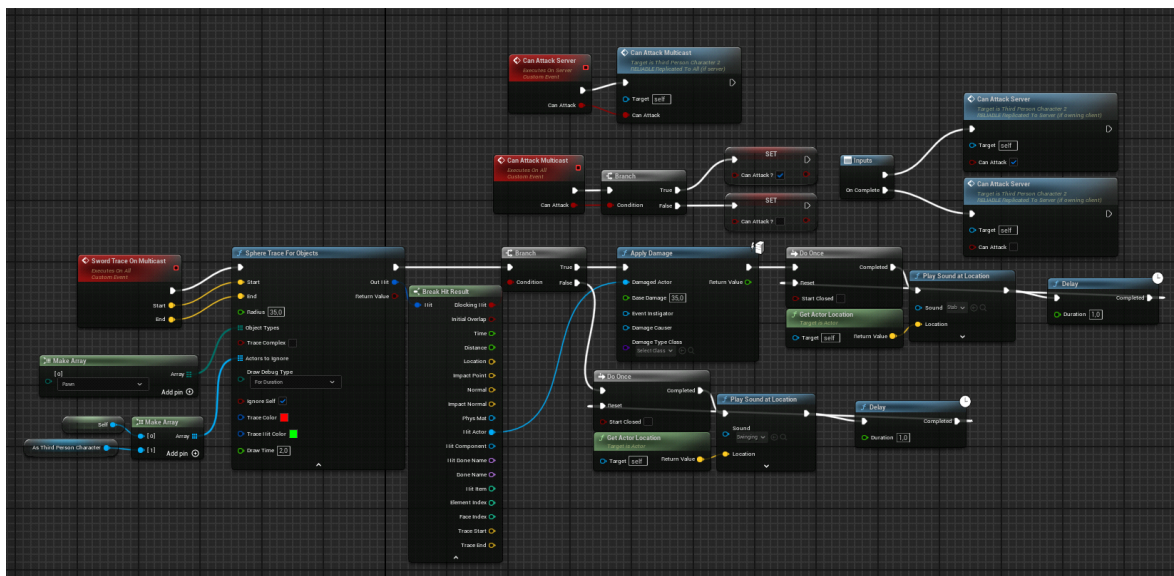
U nacrtu je osim "Line Trace"-a s kojeg ćemo kasnije još spominjati implementirano grananje s uvjetima radi li se određena šteta prilikom udarca (vrijednost 35) ili se ne radi šteta te su postavljeni parametri za zvuk uz određene stanke od nekoliko milisekundi za koje je dohvaćena potrebna lokacija lika. Animacija udarca šakom kao i ostale moraju se dodati u "Event Graph" od "Locomotion"-a kako bi se animacije sinkronizirale i skladno izvršavale.

Sama funkcionalnost napada mačem realizirana je na identičan način kao i funkcionalnost udarca šakom.



Slika 22 Animacija napada mačem igrača 1

Kako bi se animacije za oba slučaja izvršavale pravilno na obje strane (servera i gosta) implementirani su dodatni događaji za prikazivanje pravilne animacije "Can Attack Server" i "Can Attack Multicast".

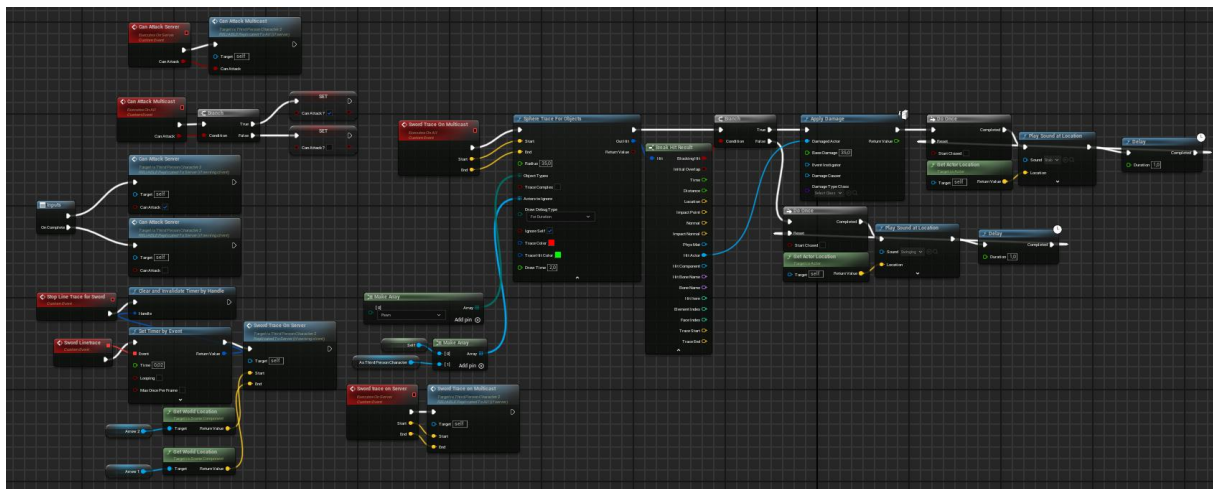


Slika 23 Nacrt implementacije napada mačem

Prilikom testiranja zamaha mača i udarca šakama testiranje se provodilo implementacijom linijskog praćenja (eng. Line Trace) pokreta objekta.



Slika 24 Linijsko praćenje udarca



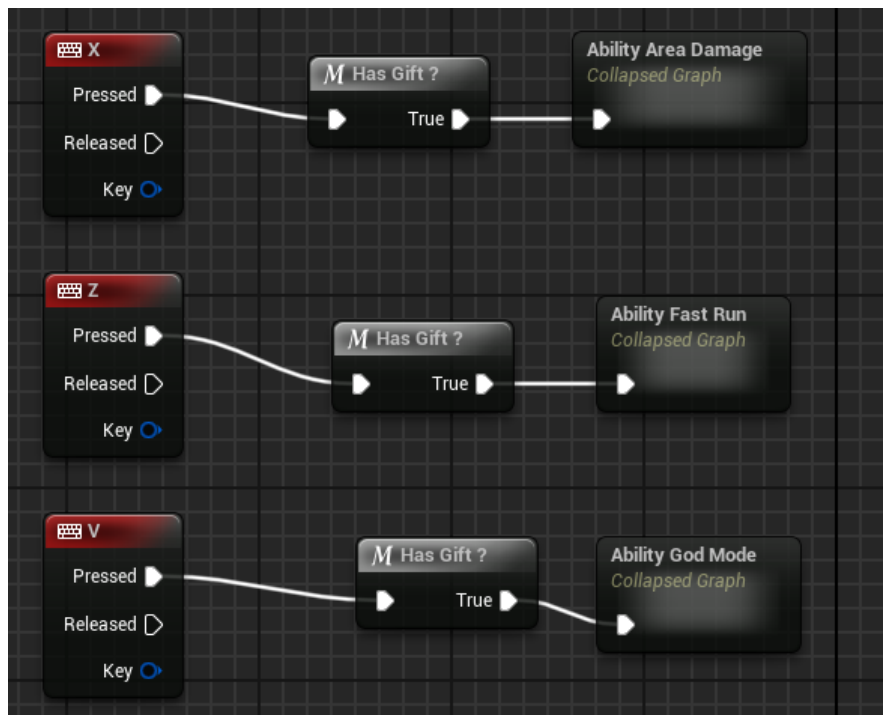
Slika 25 Implementacija linijskog praćenja mača i šake

3.2.4.1.3.3 Držanje štita

Ovu mehaniku sadrži samo igrač 1 te je implementacija vrlo jednostavna. Zajedno s ostalim oružjem ono je omogućeno nakon jedne od sekvenci priče gdje se ispituje varijabla „Gift“. Ukoliko je ona postavljena na „True“, igrač će ga moći koristiti ali u ovoj fazi videoigre samo kao ukras u svojoj lijevoj ruci.

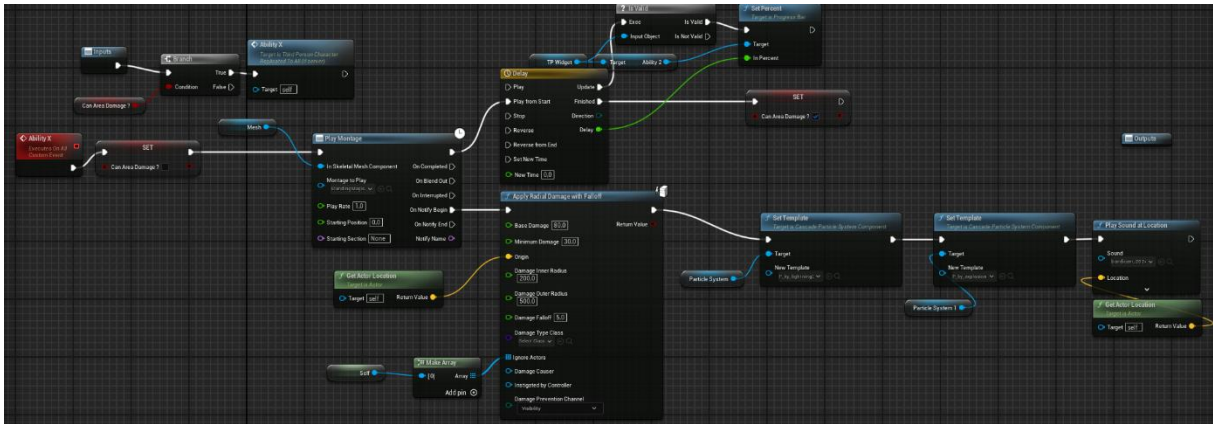
3.2.4.1.3.4 Posebne vještine

Igrač 1, zajedno s ostalim oružjem, kako vrijeme igranja odmiče dobiva 3 posebne vještine. Prva vještina omogućava mu brzo trčanje i visoko skakanje, druga mu daje gromoviti udarac u pod koji nanosi štetu te treća mu daje sposobnost besmrtnosti na nekoliko sekundi. Na slici ispod vidimo da ove vještine dobiva nakon što se ispituju čvorovi pod nazivom „Has Gift ?“ te su postavljeni inputi koji se izvršavaju ovisno o potrebi igrača.



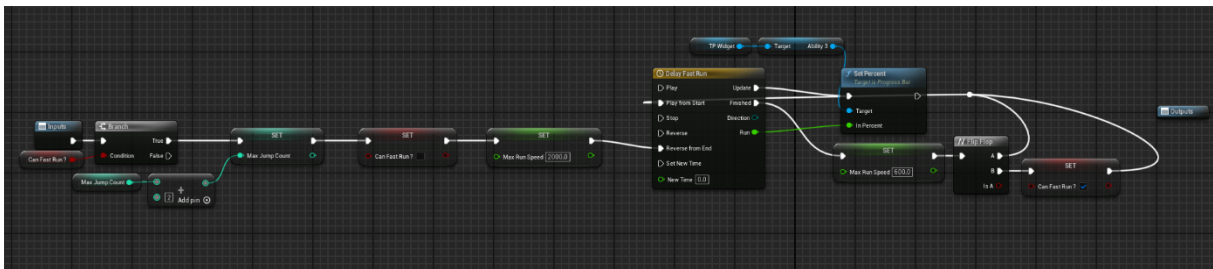
Slika 26 Inputi posebnih vještina igrača 1

U nastavku imamo nacрте sve 3 posebne vještine. Za animaciju vještine udarca u pod bilo je potrebno pravilno implementirati setter i pokrenuti montažu te postaviti određeni radius do kuda će se određena šteta neprijateljima raditi, postaviti određenu pauzu između intervala u postotcima za izvršavanje vještine i dodati dodatne vizualne efekte i zvuk prilikom reproduciranja.



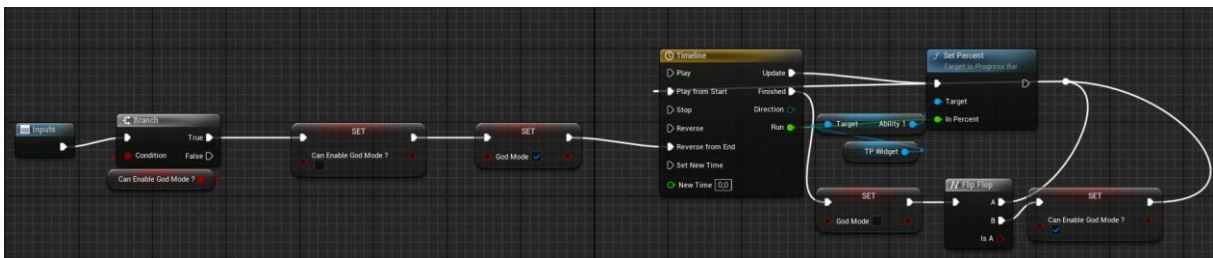
Slika 27 Posebna vještina udarca u pod

Kod vještine brzog trčanja dodana je opcija uz pomoć settera na maksimalno visoko skakanje od dva puta čija se logika prvo implementirala i pozvala te je postavljena maksimalna brzina prilikom izvršavanja ove vještine na 2000. Dodana je pauza između izvođenja iste vještine dva put te se smanjuje brzina trčanja na standard čija vrijednost iznosi 600.



Slika 28 Posebna vještina brzog trčanja

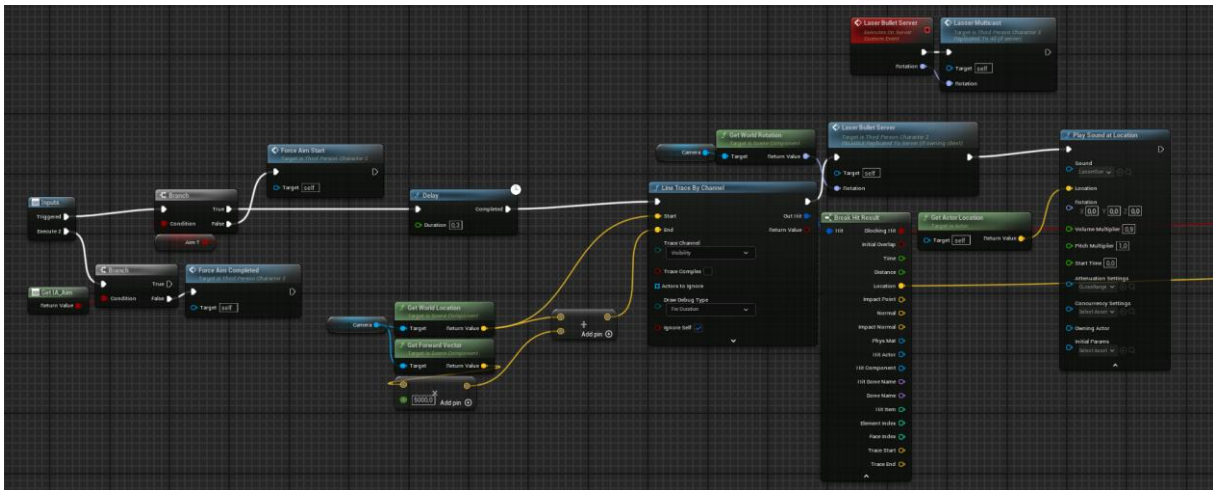
Kod posljednje vještine također se ispituju uvjeti i postavlja se mogućnost izvršenja vještine samo jedan put u određenom razdoblju.



Slika 29 Posebna vještina besmrtnosti

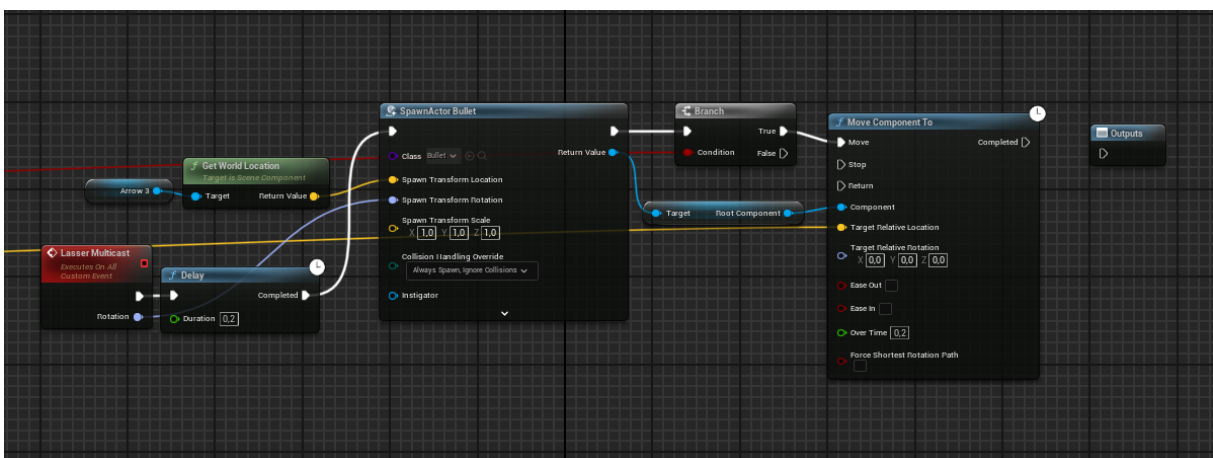
3.2.4.1.3.5 Napad laserom

Ova vještina moguća je samo za igrača 2 čiji lik ima poseban alat na ruci koji služi za pucanje lasera. Prije implementacije pucanja, na slikama ispod imamo nacrt u koji smo dodali novu kameru. Putem dohvaćanja scene lokacije svijeta u stvarnom vremenu kamera omogućuje igračima da ako pritisnu desni klik miša fokusiraju pogled lika nakon čega mogu izvršiti napad laserom.

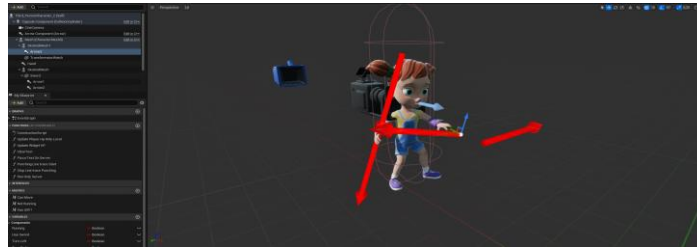


Slika 30 Nacrt lasera 1

Također je prilikom implementacije bilo potrebno dodati vizualni objekt (*eng. Actor*), u ovom slučaju pod nazivom „Bullet“ koji se ručno kreirao te se ispaljuje u obliku plave laserske zrake kao i zvučni efekti te mogućnost vidljivosti ovih efekata za oba igrača. Potrebno je korektno dodati lokaciju odakle će se laserska zraka pojaviti, primjerice „Get World Location“ i dodati za metu u ovom slučaju „Arrow 3“, odnosno strelicu koja izlazi iz alata lika.



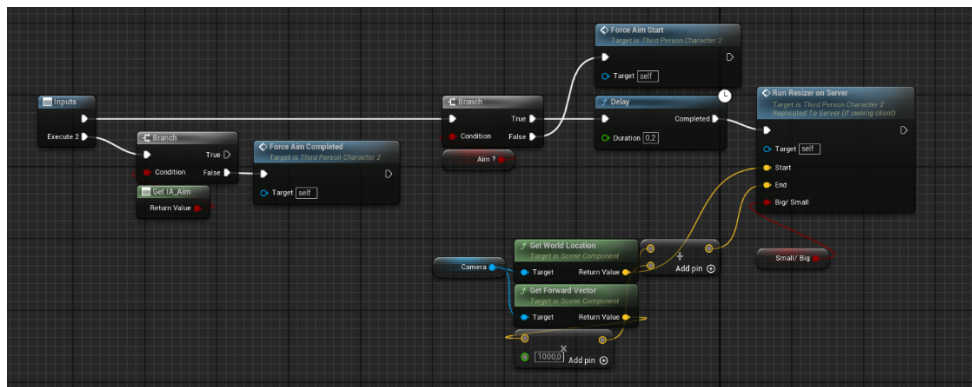
Slika 31 Nacrt lasera 2



Slika 32 Implementacija strelice lasera

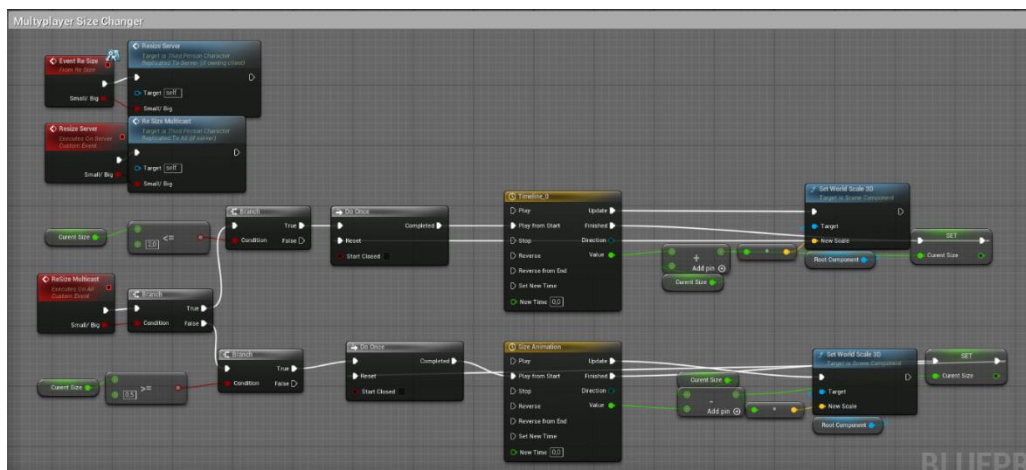
3.2.4.1.3.6 Smanjivanje i uveličavanje likova

Nakon fokusiranja kamere kao i kod pucanja lasera, implementirali smo mogućnost smanjivanja i uveličavanja likova na mapi vidljivo za oba lika.



Slika 33 Nacrt fokusa smanjenja ili uveličavanja likova za server

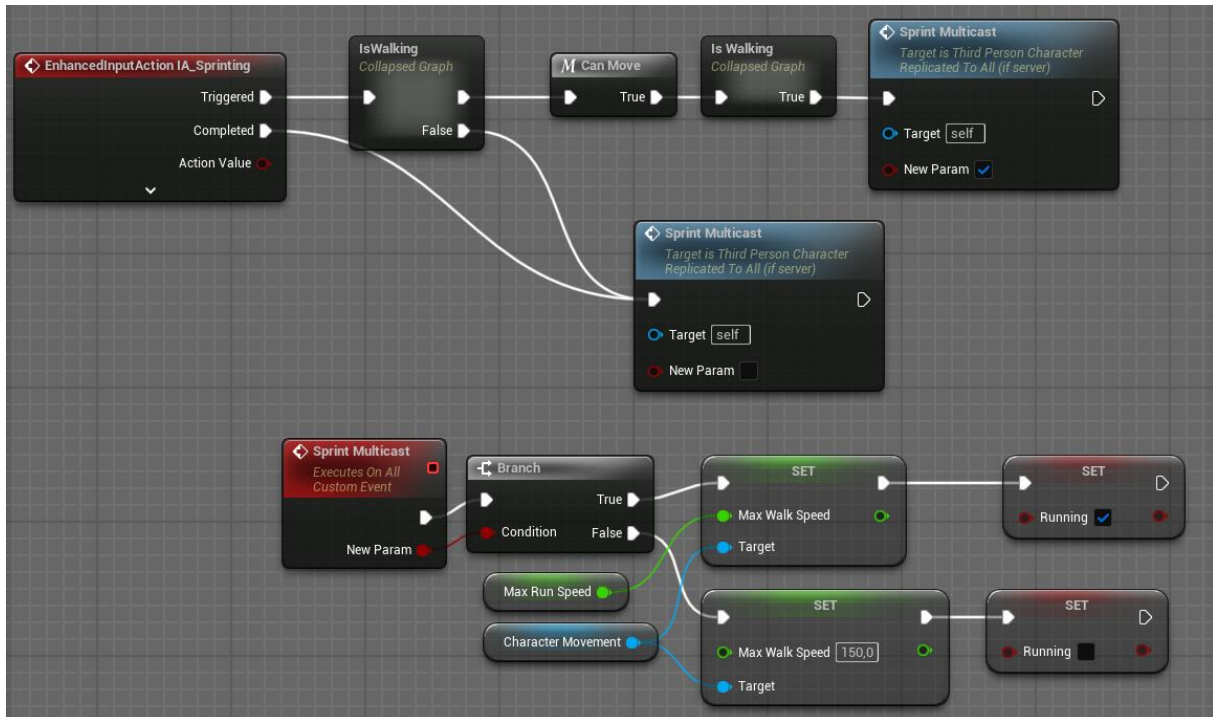
Proces implementacije treba dovršiti da je aktivnost vidljiva i za gosta. Dodani su uvjeti za varijable uz pomoć grananja ovisno o tome želimo li smanjiti ili povećati lika. Aktivnost je omogućena dodavanjem „Timeline“ nacrtu kojim se određuje vrijeme trajanja ove aktivnosti te se postavlja 3D skala svijeta za ažuriranje ovog događaja na određenoj lokaciji.



Slika 34 Nacrt smanjenja i povećanja lika za server i gosta

3.2.4.1.3.7 Trčanje

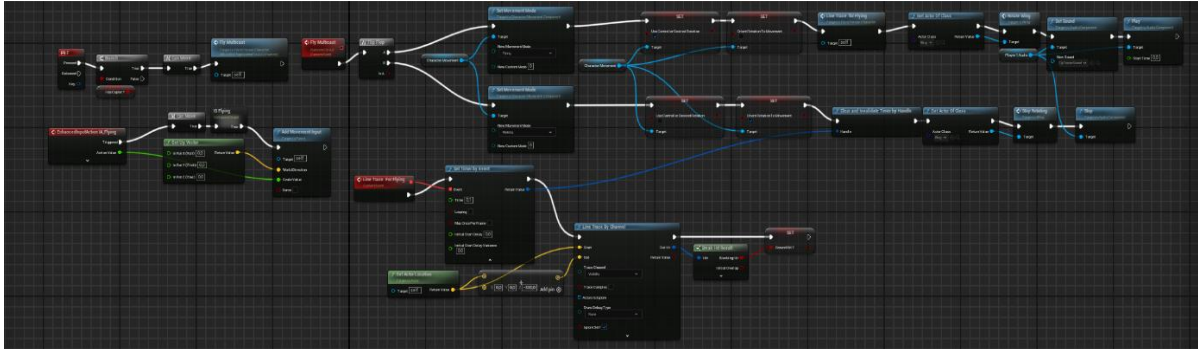
Pritiskom pravog inputa pokrenut će se „EnhancedInputAction IA_Sprinting“ te se prilikom implementacije također pazilo na prikazivanje trčanja pravilno na obje igrачke strane.



Slika 35 Nacrt inputa trčanja

3.2.4.1.3.8 Letenje

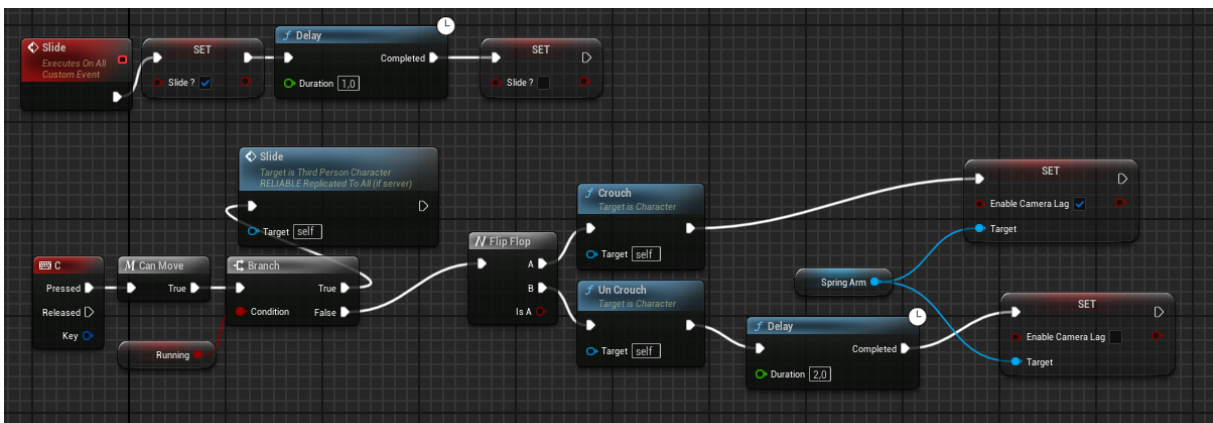
Kao i kod prethodnih mehanika, provjerava se pokretljivost lika uz pomoć ućahurivanja logike i korištenjem ćvora (eng. Macro) unutar nacrtu pod nazivom „Can Move“ te se dodaju vektori za svijet. Makro dopušta definiranje funkcija koje se mogu izvršiti više puta u različitim dijelovima nacrtu. Postavlja se odrećeni naćin rada (letenje ili hodaenje) te ovisno o tome postavljen je setter za propeler i njegovu rotaciju koja je implementirana unutar „Mesh“-a objekta. Na kraju se dodaje odrećeni zvuk ili se zaustavlja animacija letenja. Kao i za sve aktivnosti, tako je i ova vidljiva oba igraća.



Slika 36 Nacrt inputa letenja

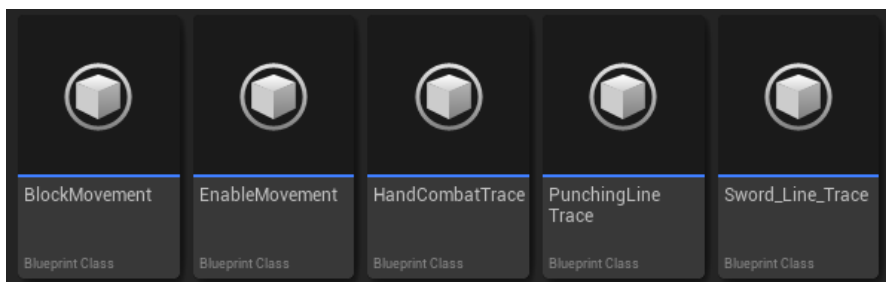
3.2.4.1.3.9 Čučanj i proklizavanje po podu

Prilikom čučnja implementacija je standardna te se provjeravaju uvjeti, no prilikom proklizavanja po podu potrebno je bilo dodati uvjet da lik trči a zatim pritisne tipku „C“.



Slika 37 Nacrt čučnja i proklizavanja

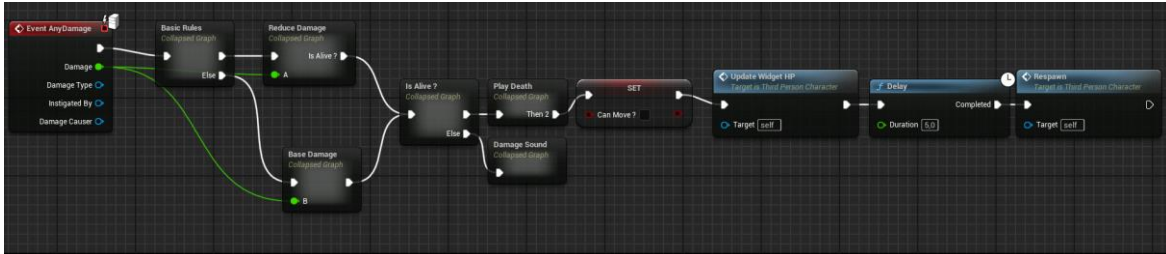
Prilikom ovakvih mehanika moguća implementirane su neke od komponenti „Animation Notify State“-ova kako bi se ograničilo likovima prilikom pokretanja animacije dodatno kretanje unaprijed ili omogućilo. Također neke klase stanja poslužile su za testiranje kao linije udarca.



Slika 38 Animation Notify State

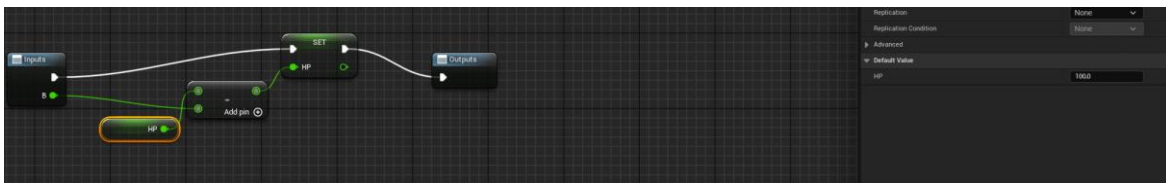
3.2.4.1.3.10 Ranjivost i zdravlje

Glavni događaj koji smo dodali u nacrt za situaciju ranjivosti i zdravlja je "Event AnyDamage". Dodane su funkcionalnosti za provjeru određenih pravila, smanjivanje zdravlja, postavljanje uvjeta smrti te njihovo ažuriranje unutar "Widget"-a igrača.



Slika 39 Događaj "AnyDamage"

„Base Damage" sadrži nacrt za oduzimanje zdravlja lika čija vrijednost zdravlja vidimo da je postavljena na 100 na slici ispod.



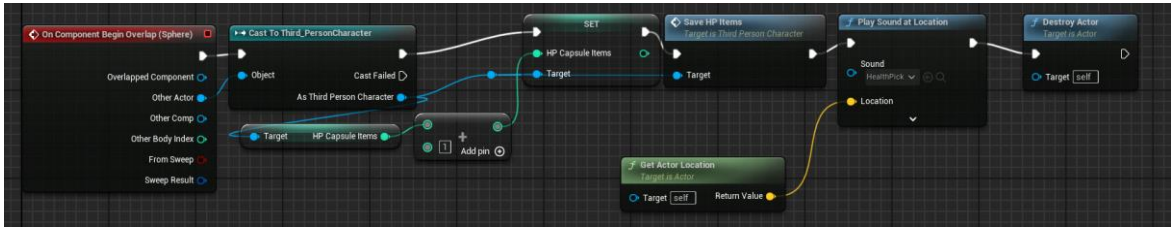
Slika 40 Vrijednost zdravlja

Kako bi igrač izbjegao smrt, može pokupiti srce koje ga ozdravi i daje mu dodatnu snagu.



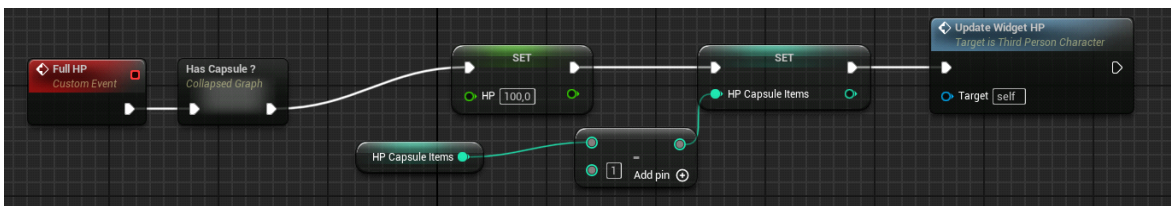
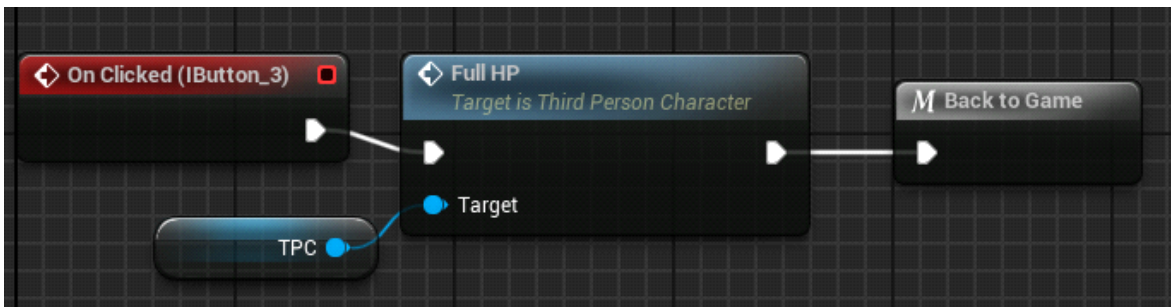
Slika 41 Entitet srce

Kada ga dotakne, pokupi ga te se unutar kapsule s kolekcijom doda jedno srce koje igrač može iskoristiti kako bi vratio zdravlje na 100%.



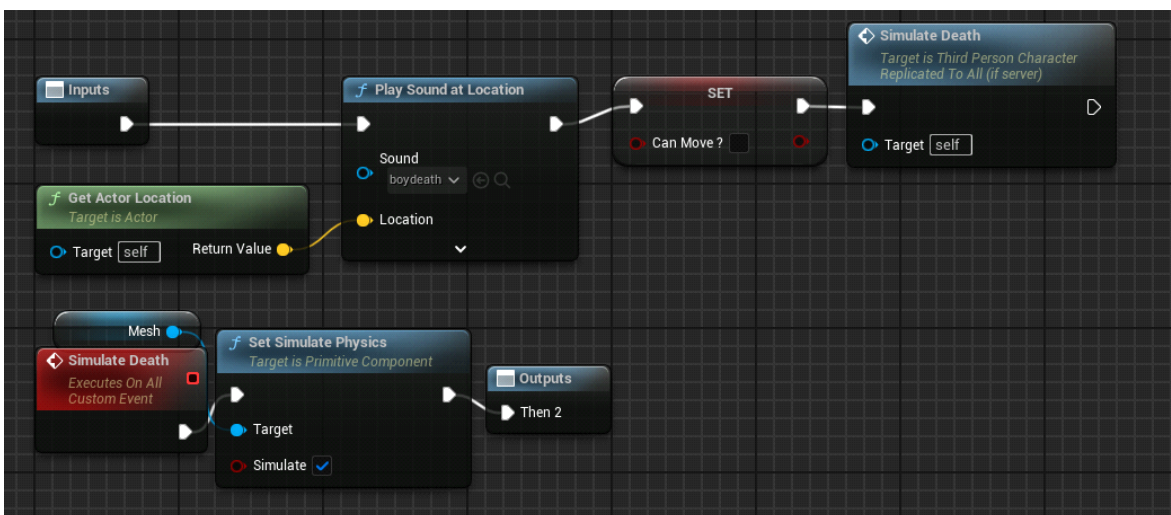
Slika 42 Sfera srca

Slike ispod demonstriraju nacрте za klik na srce kolekcije te implementaciju funkcionalnosti postavljanja igračeva zdravlja na 100%.



3.2.4.1.3.11 Smrt

Do smrti se jedino može doći ako glavni likovi ili neprijatelji izgube cijeli postotak života, odnosno indikator života dođe na nulu. Tada će se u nacrtu igrača 1 i igrača 2 ili neprijatelja aktivirati se događaj "Play Death".



Slika 43 Nacrt simulacije umiranja

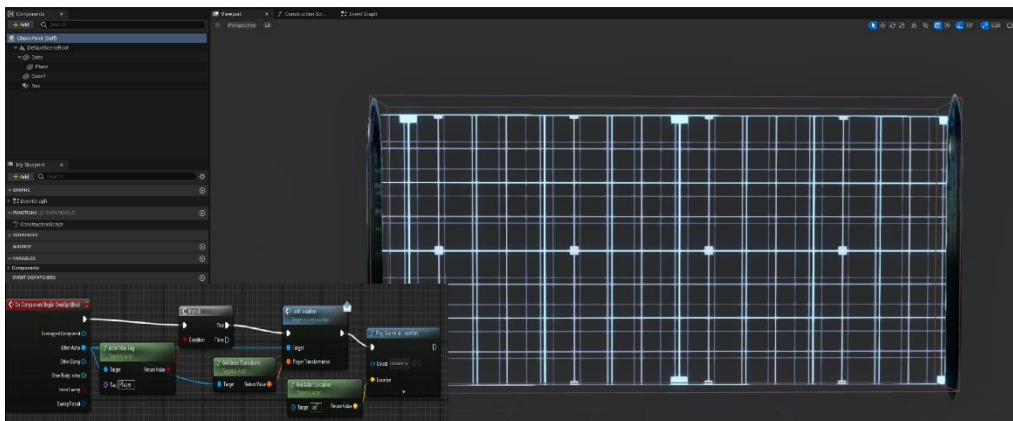
3.2.4.1.3.12 Oživljavanje

Imamo dvije vrste oživljavanja, odnosno mjesta oživljavanja likova. Po standardu oživljavanje likova i njihovo ponovno stvaranje u svijetu prikazano je na nacrtu ispod. Prilikom oživljavanja lik se teleportira na mjesto gdje je i umro te se njegova kapsula i „Mesh“ koriste kao mete. Prilikom oživljavanja postavlja se mogućnost kretanja lika ponovo uz vraćanje zdravlja na 100% što je vidljivo u indikatoru zdravlja igrača.



Slika 44 Oživljavanje i teleportacija igrača

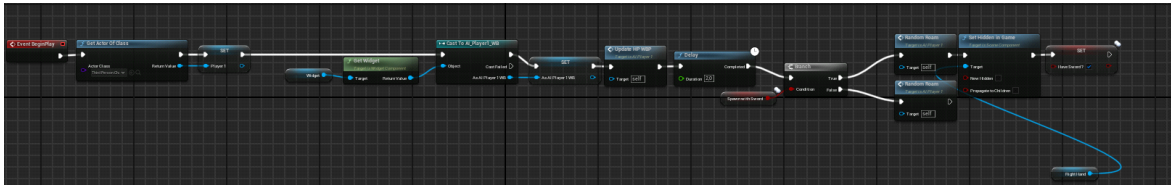
Za potrebe platformerskog svijeta kreiran je objekt (*eng. Actor*) čijim prolaskom kroz njega igrač se nakon smrti ponovno stvore pored objekta kroz koji su zadnji fizički prošli (*eng. Checkpoint*).



Slika 45 Checkpoint

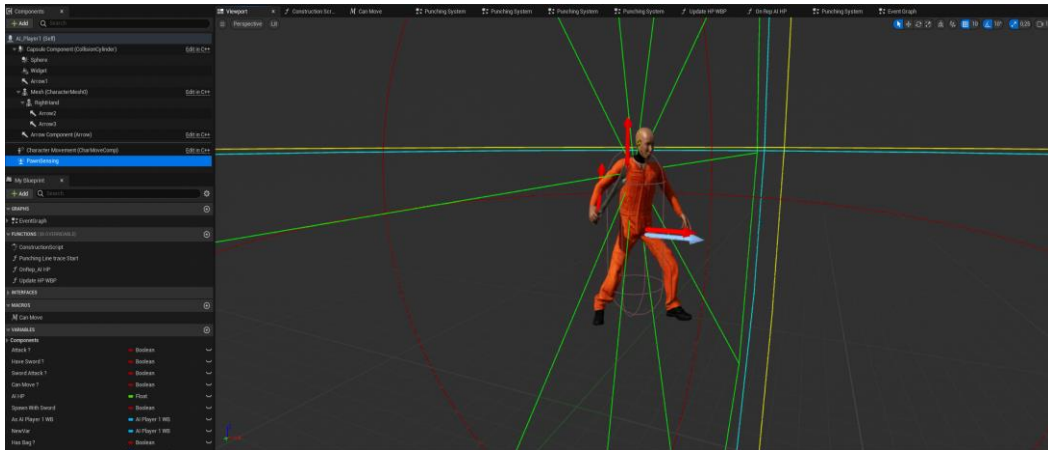
3.2.4.1.4 Kreiranje neprijateljskih funkcionalnosti

Neprijatelj isto kao i glavni likovi koristi puno sličnih funkcionalnosti, no zbog preglednosti njihovi nacrti odvojeni su te smo neprijateljev nacrt za implementaciju glavnih funkcionalnosti nazvali "AI_Player1". Prilikom implementacije izbacili smo input događaje koji su nam recimo potrebni kod glavnih igrača no ostavili smo osobine poput napada mačem, oduzimanje zdravlja i sl. Ove funkcionalnosti spojene su s "Locomotion"-om neprijatelja pod nazivom "AIPlayer1ABP".

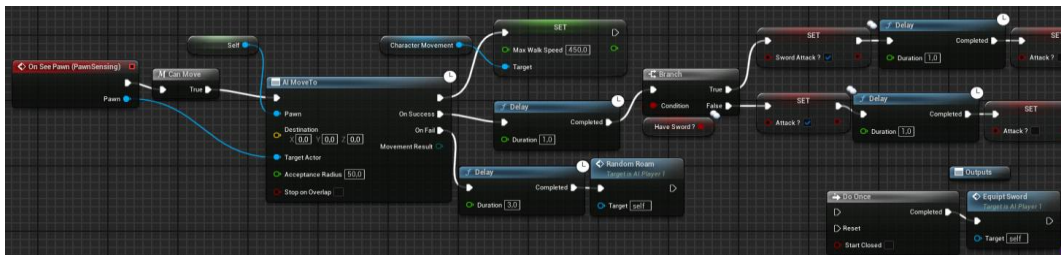


Slika 46 Događaj "EventBegin" za neprijatelja

Implementirana je i postavka "PawnSetting" koja omogućava neprijatelju da putem senzora (čiji parametri udaljenosti su definirani) može detektirati igrača te ga pratiti i napasti.

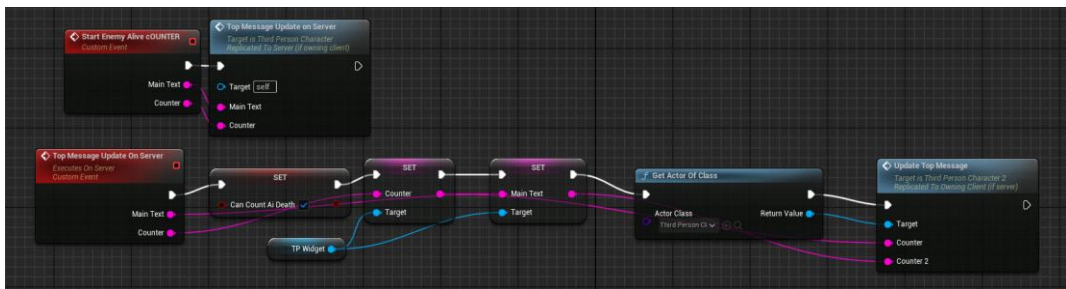


Slika 47 "Pawn Sensing viewport"



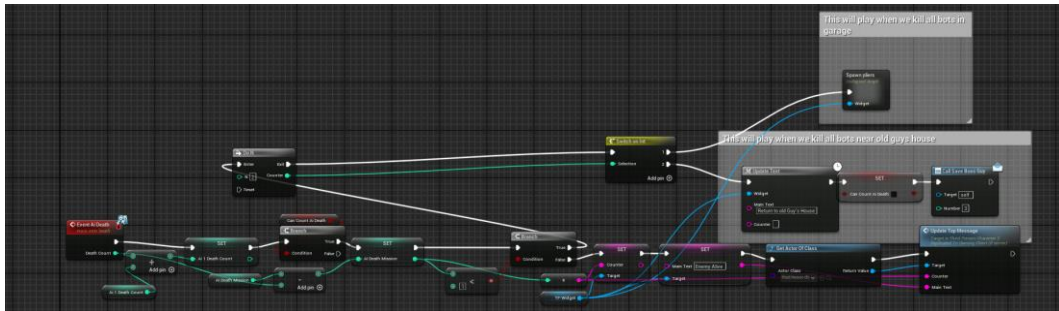
Slika 48 Nacrtna implementacija "PawnSensing"-a

Za navođenje igrača o broju preostalih živih neprijatelja implementiran je sustav odbrojavanja broja živih neprijatelja gdje se informacije ažuriraju na serveru.



Slika 49 Nacrtna brojača preživjelih neprijatelja na serveru

Na ovom nacrtu imamo primjer odbrojanja živih neprijatelja za jednu od misija videoigre.

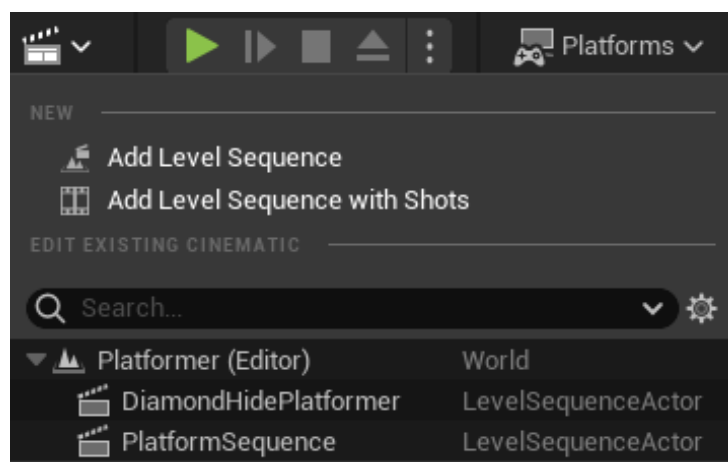


Slika 50 Nacrt za brojač živih neprijatelja

Prilikom stvaranja neprijatelja u svijetu, stvaraju se zajedno s mačem u ruci te dobivaju "Widget" zdravlja iznad svoje glave.

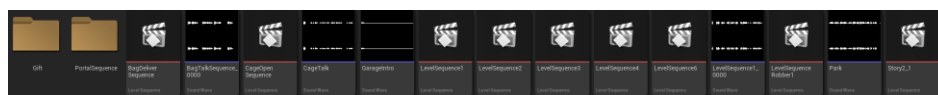
3.2.4.1.5 Animacijske video sekvence

Kako bi videoigra bila zanimljivija za igrače, kreirane su animacijske video sekvence putem koje igrači prate određeni tok igre (dobivaju informacije) i dobivaju iskustvo igranja uz priču. Sekvence su realizirane u „Animation“ uređivaču. Sam proces započinje kreiranjem nove sekvence.



Slika 51 Dodavanje nove sekvence

Za ovaj diplomski rad kreirano je nekoliko animacijskih video sekvenca.



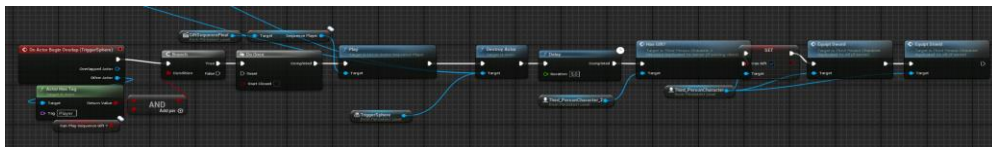
Slika 52 Sekvence projekta

Svaka sekvenca sadrži nekoliko kadrova, outpute zvuka, vizualne efekte poput "Fade" efekta za stvaranje ambijenta toka priče te je za svakom "Mesh"-u bilo potrebno dodati odgovarajuće animacije te postaviti ključeve lokacija animacije likova i objekata.



Slika 53 Primjer kreiranja video animacijske sekvence s parametrima

Svaka sekvenca pokrenula bi se ovisno o zadacima koje su igrači odradili te ih igra navodi da sa svojim likovima dođu u određenu nevidljivu sferu (njima je prikazana u obliku navigacijskog kruga) gdje se nakon toga aktivira sekvenca koja je primjerena za tok igre. U ovom slučaju nakon što se sekvenca završi pojedini igrač dobit će svoje oružje.



Slika 54 Primjer nacrtu za pokretanje video sekvence

3.2.4.1.6 Ostale funkcionalnosti i značajke

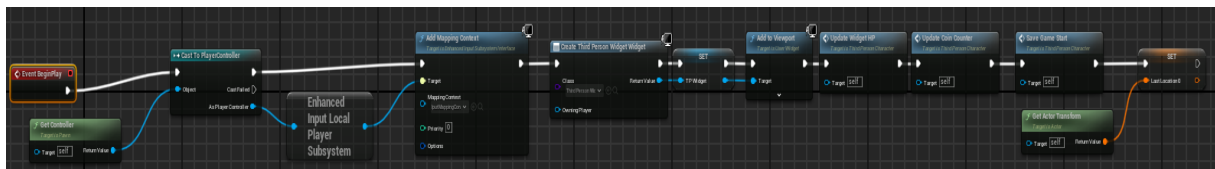
3.2.4.1.6.1 Kreiranje „Widgeta“ igrača

Kako bi igrači bili informirani o zdravlju njihovog lika, broju novčića koji posjeduju ili pak vještine koje posjeduju dizajnirani su „Widgeti“ za svog od igrača.



Slika 55 „Widgeti“ igrača 1 i 2

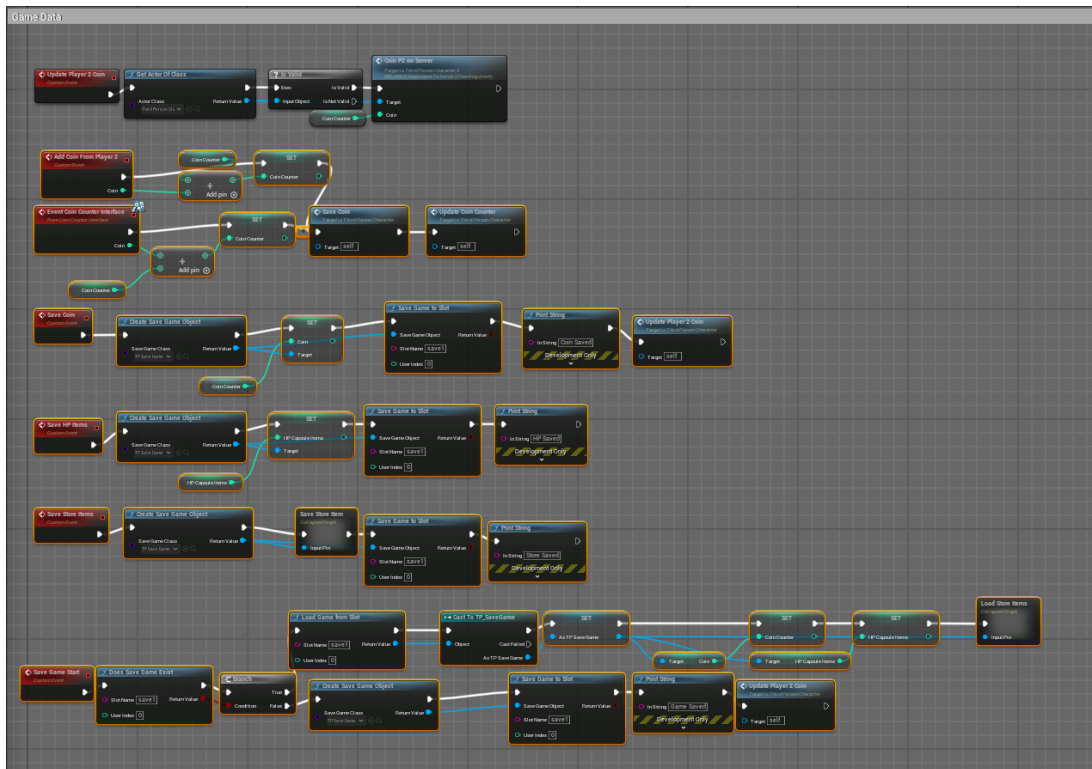
Prilikom pokretanja igre unutar „Event BeginPlay“ postavili smo opciju da se ovi Widgeti za svakog igrača automatski prikažu te da se pojedine komponente sesije spremaju.



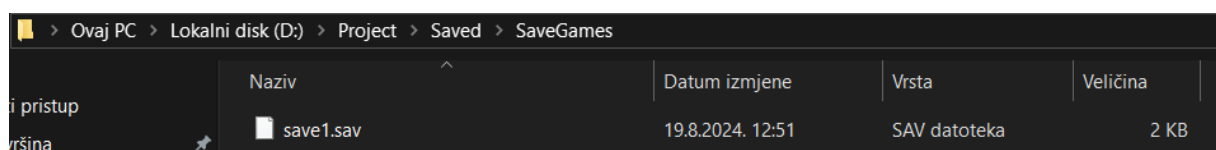
Slika 56 Događaj "Event BeginPlay"

3.2.4.1.6.2 Spremanje podataka

Vrlo važna komponenta igre je i samo spremanje sesije igranja videoigre. Spremit će se prikupljeni novčići, prikupljena srca za ozdravljenje kao i oružje koje je do tad kupljeno. Ove komponente spremaju se u datoteku pod nazivom „save1“ koja se nalazi u projektnoj datoteci.



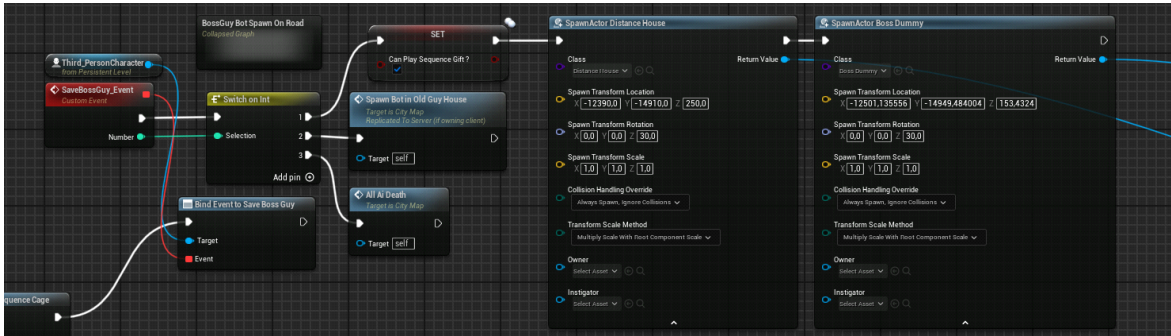
Slika 57 Prilagođeni događaji za spremanje prikupljenog sadržaja videoigre



Slika 58 Putanja do datoteke "save1"

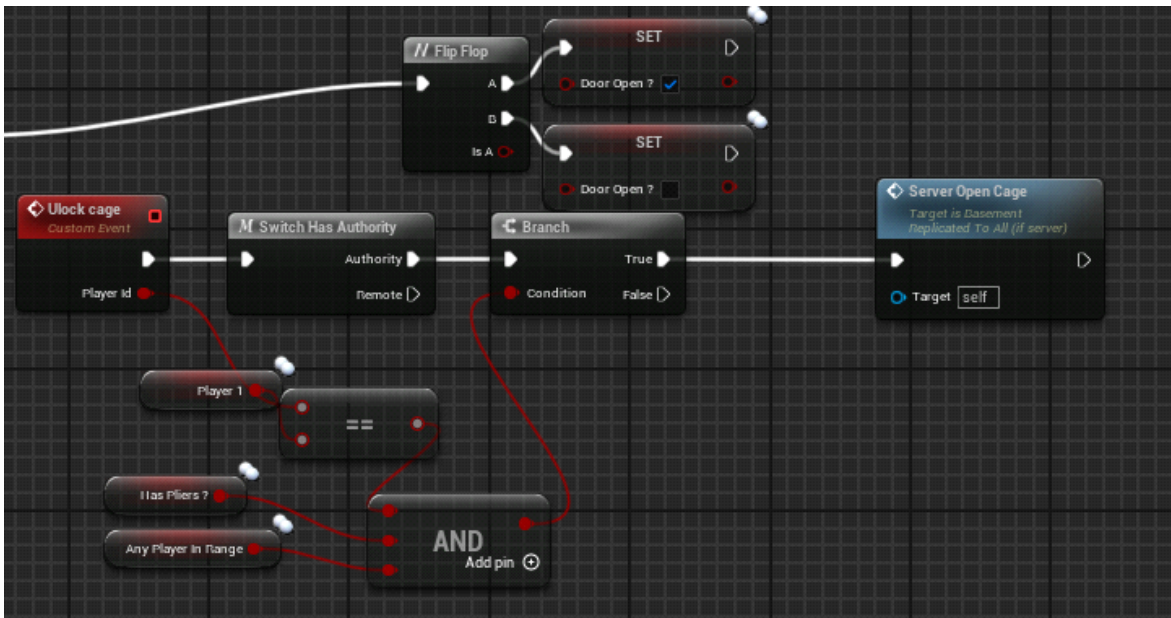
3.2.4.1.6.3 Misija spašavanja starca

Za jednu od misija bilo je potrebno implementirati niz događaja uključujući kada će se stvoriti neprijatelji, kada će se ponovno pojaviti starac i likovi igrača, te što će se pokrenuti nakon što su svi neprijatelji mrtvi.



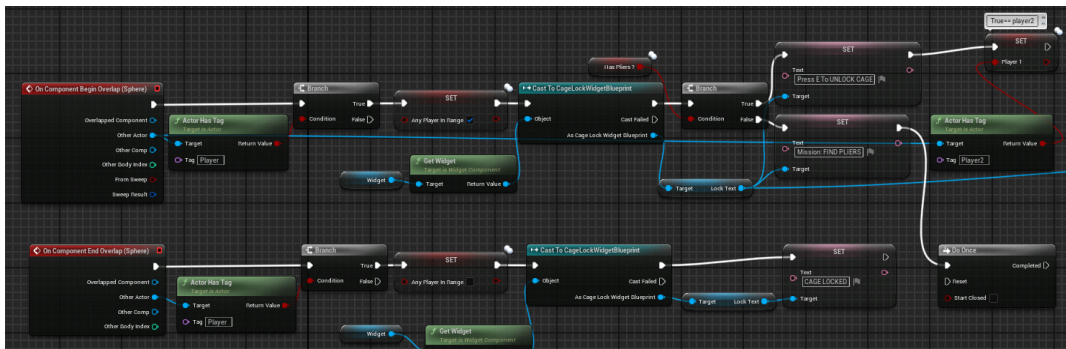
Slika 59 Nacrt implementacije Switch-a za događaje

Kod spašavanja starca iz kaveza bilo potrebno implementirati logiku primjerice kada je moguće otključati kavez. Pa tako u ovom slučaju potrebno je bilo pokupiti prvo kliješta te nakon toga otključati kavez.



Slika 60 Nacrt logike otključavanja kaveza

Nacrt ispod prikazuje ograničenja koja se nalaze na kavezu te putem običnog teksta navode igrača što može a što ne uraditi.



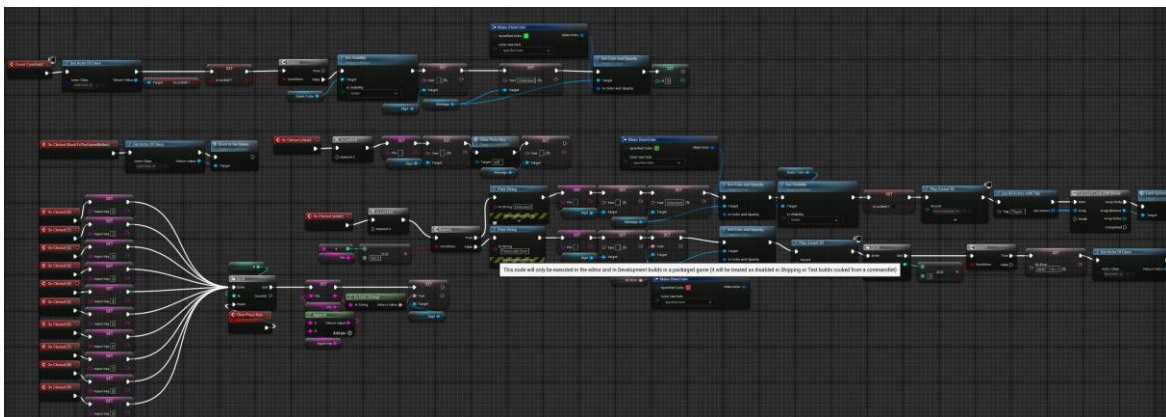
Slika 61 Nacrt ograničenja kaveza

Kako bi se došlo do neprijatelja i starca igrač implementiran je digitalni sustav lokota garaže koju igrač mora dešifrirati.



Slika 62 Dizajn izrade sustava digitalnog lokota

Implementiran je niz značajki, uključujući ispis svakog broja prilikom pritiska na određeni gumb, točan PIN digitalnog lokota, ispis ispravnosti unosa PIN-a, te postavljanje zelene ili crvene boje lokota ovisno o uspješnosti.



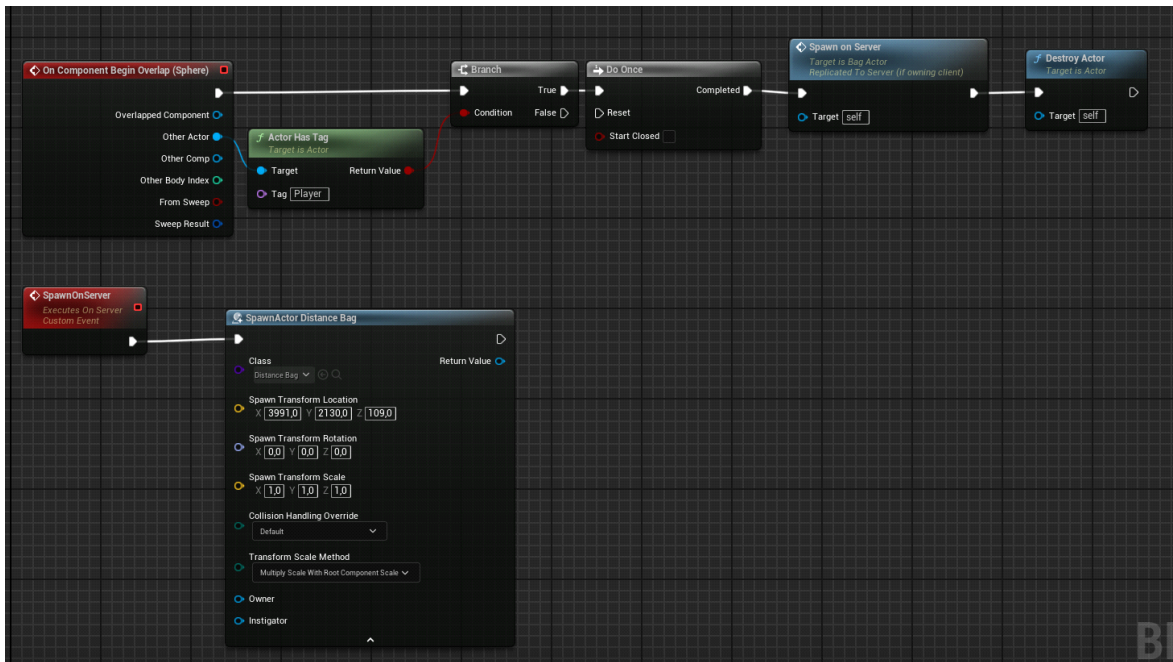
Slika 63 Nacrt pravila lokota

3.2.4.1.6.4 Misija hvatanja lopova

U nacrtu na slici 64 implementiran je setter koji upućuje da jedan od neprijatelja ima torbicu. Ona se nakon što neprijatelj umre stvori te ju igrač može pokupiti i vratiti vlasnici.



Slika 64 Nacrt implementacije povezivanja torbice s neprijateljem



Slika 65 Nacrt nastajanja torbice na određenoj lokaciji i njezina sfera

Za potrebe ove misije dizajniran je neprijatelj s torbicom u ruci te je korišten prilikom izvođenja animacijske video sekvence.



Slika 66 Neprijatelj s ukradenom torbicom

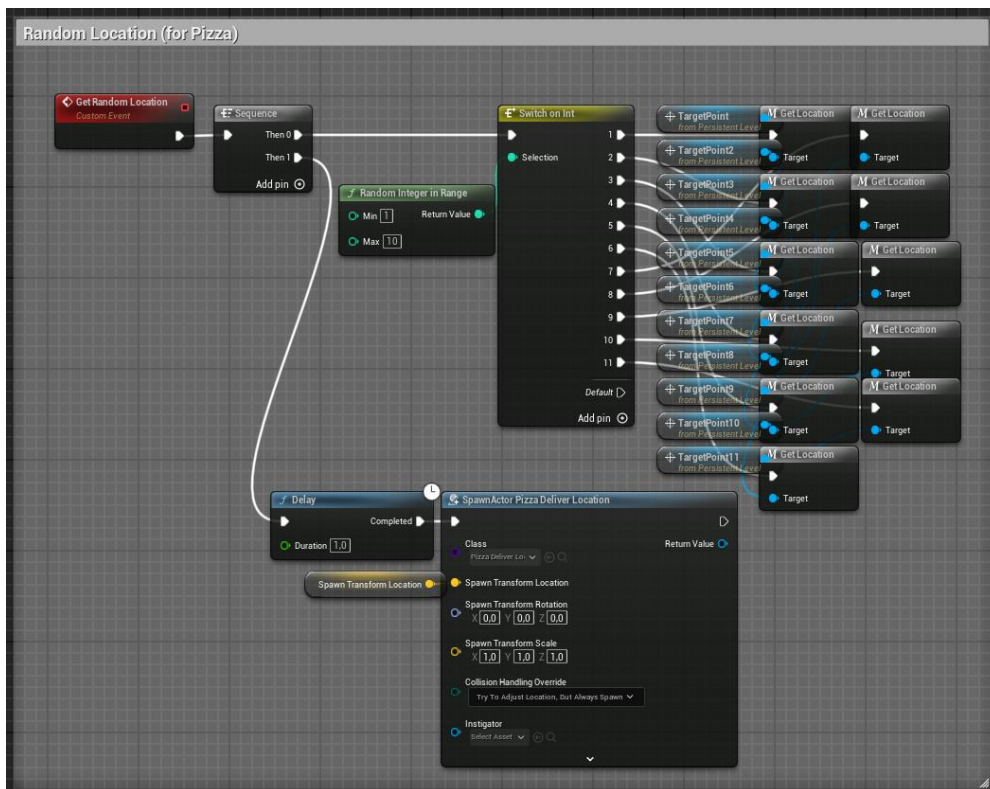
3.2.4.1.6.5 Sporedna misija dostave pizze

Za igrača je dodana dodatna misija s kojom imaju mogućnost prikupiti novčiće brže i efikasnije tako da obavljaju posao dostave pizze na različite lokacije. Sve što je potrebno je da odu u restoran i prihvate posao.



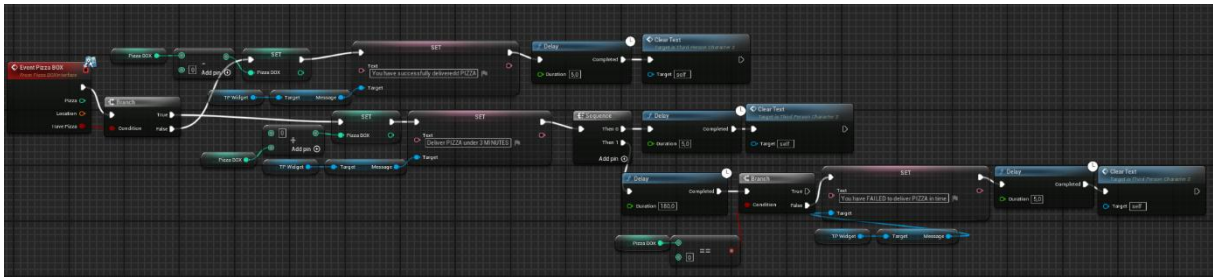
Slika 67 Restoran za dostavu pizze

Unutar nacrt „CityMap“ svijeta dodan je prilagođen događaj koji se aktivira kada igrač prihvati ponudu dostave pizze. Nasumično se odredi lokacija na kojoj se stvori ikona pizze i kruga gdje igrač mora stati kako bi dostava pizze unutar 3 minute bila uspješna.



Slika 68 Nacrt iz "CityMap" svijeta za nasumično postavljanje dostave pizze

Dodan je događaj za instancu dostave na način da se određeni tekst ispiše ovisno o uspješnosti dostave u zadanom vremenu. Igrač je nagrađen s 5 novčića ukoliko je zadatak izvršen.



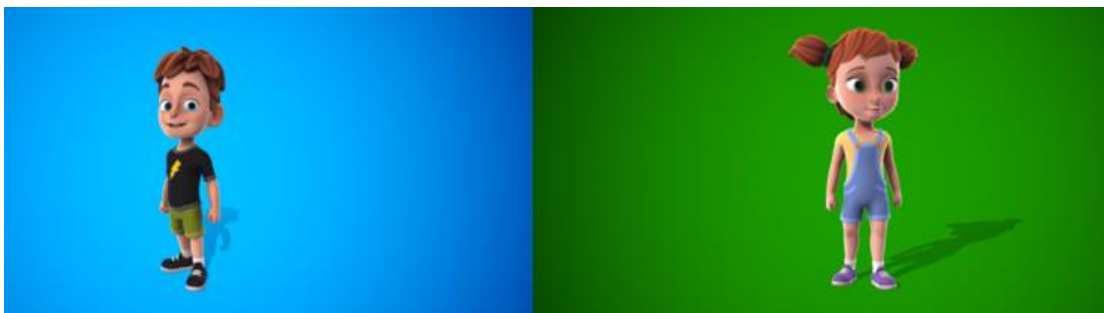
Slika 69 Nacrt događaja "Event Pizza BOX"

3.2.4.2 Odabir i izrada 3D modela i animacije

Za potrebe ove videoigre, ideja je bila kreirati videoigru s likovima koji bi bili crtanog stila (eng. Cartoon style) te bi davali određeni dojam zabave, interaktivnosti i pristupačnosti i za igrače manjeg uzrasta unutar svijeta u kojem bi ih kontrolirali. Kroz čitav projekt, nastoji se rabiti što više prikladnijih pokreta i animacija za likove kako bi igrači dobili još jednu dozu iskustva.

3.2.4.2.1 3D modeli likova

Zbog određene želje za originalnosti u izradi projekta, 3D modeli glavnih likova kupljeni su putem Epic Games-ove web trgovine koja nudi veliki broj 3D modela, efekata i sl. 3D modeli objekata poput kuća, dječjeg igrališta, garaže, restorana, kliješta i ostalih većih ili manjih modela objekata preuzeti su kao gotovi proizvod koji je dostupan i besplatan svima.



Slika 70 3D dizajn glavnih likova

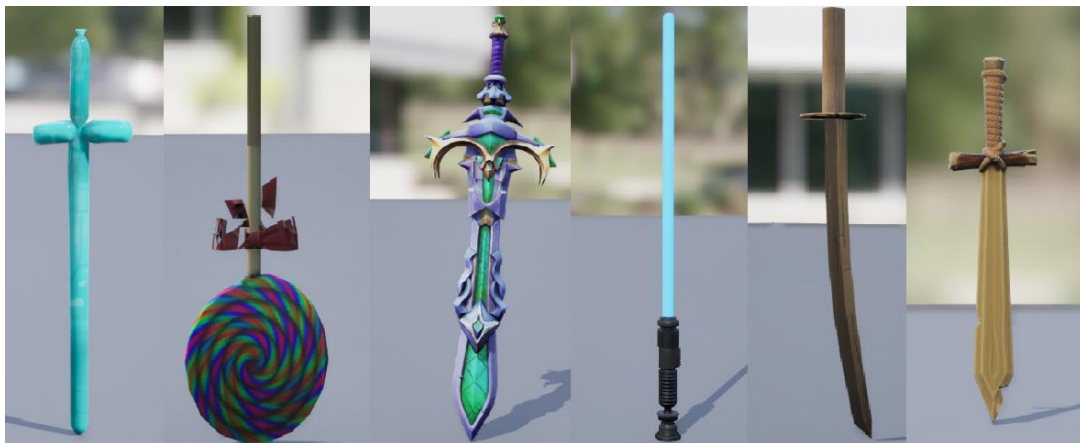
Sporedni likovi preuzeti su na javno dostupnoj web stranici „Mixamo“ prvenstveno stvorenoj za generiranje potrebnih animacija koje se zatim implementiraju na 3D modele likova unutar samog softvera:



Slika 71 3D dizajn sporednih likova

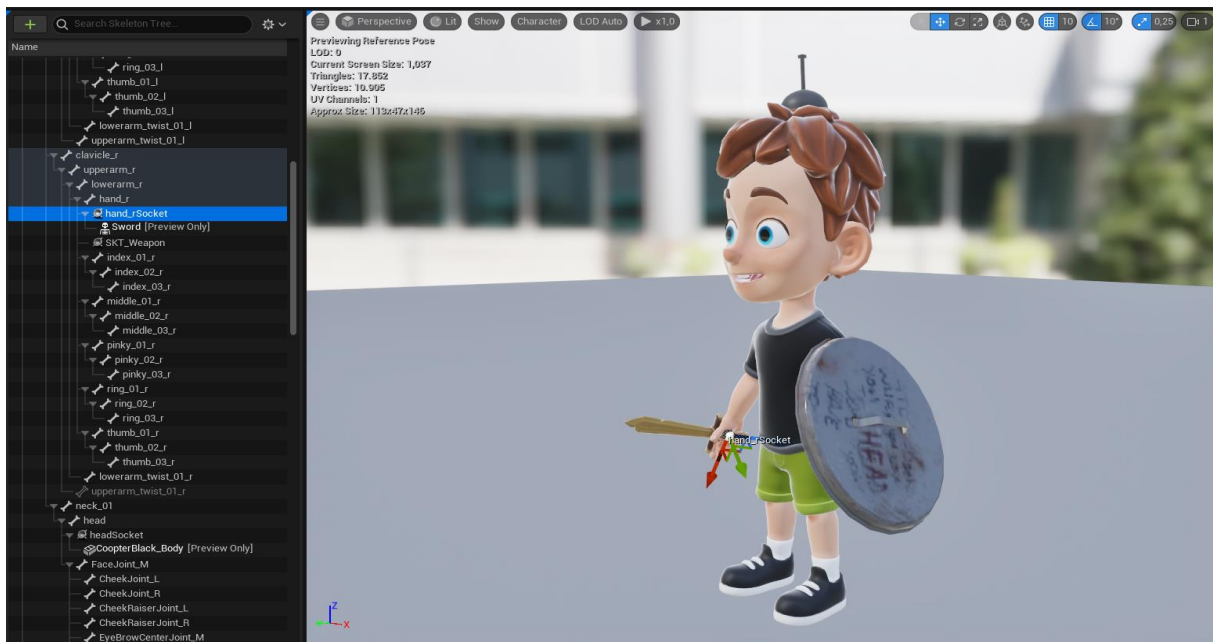
3.2.4.2.2 3D modeli oružja

Svaki model oružja, općenito, stavlja se na ruke 3D modela na način da se ode u pojedini kostur, označi lijeva ili desna šaka te se doda „Mesh“ oružja. Po potrebi je moguće rotirati i pomicati oružje kako bi držanje objekta izgledalo što realističnije.

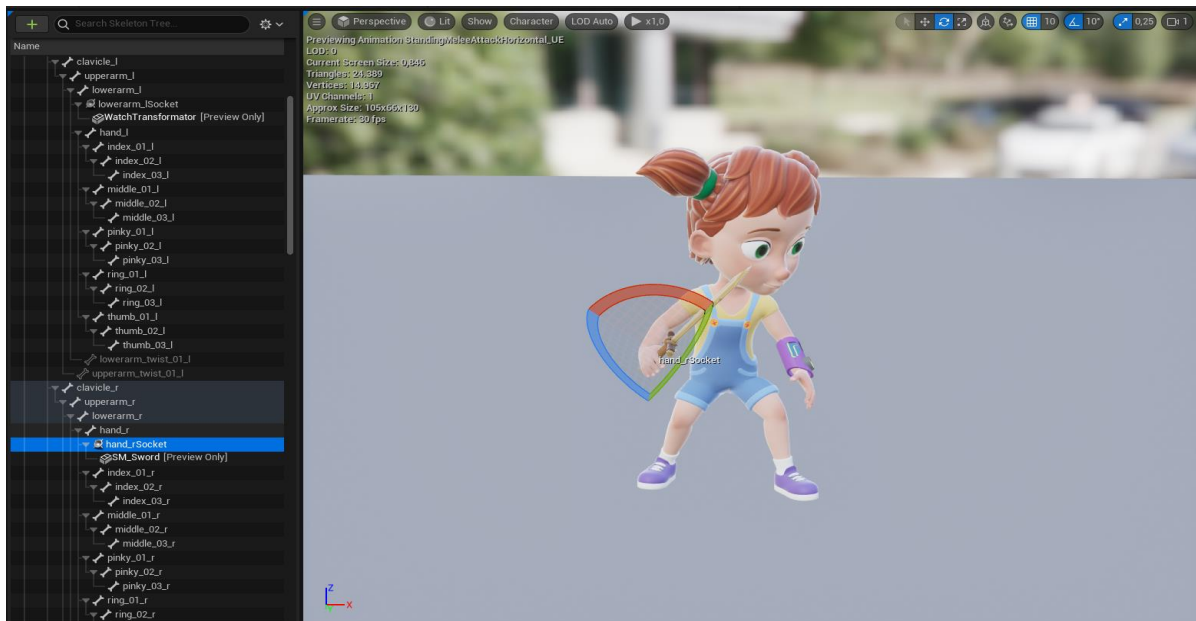


Slika 72 3D modeli oružja

3.2.4.2.3 3D modeli igrača i neprijatelja



Slika 73 Štit, mač i propeler igrača 1



Slika 74 Mač i laserski alat igrača 2

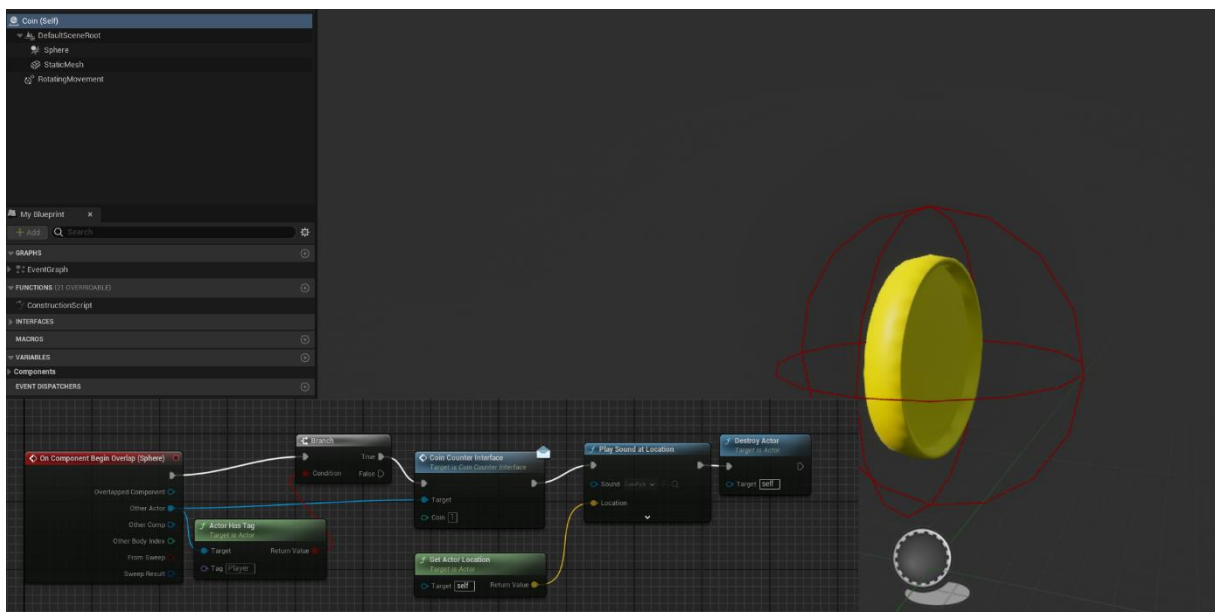
Ono što je također važno prilikom naoružavanja likova je postaviti početak drške mača i kraj oštrice mača kako bi odredili koliziju. Također prilikom testiranja udaraca i zamaha mačem ili šaka moguće je postaviti trag linije (*eng. line trace*) što nam omogućava da znamo kada je oružje dotaknulo a kada ne određeni objekt.



Slika 75 Postavljanje točke kolizije oružja neprijatelja

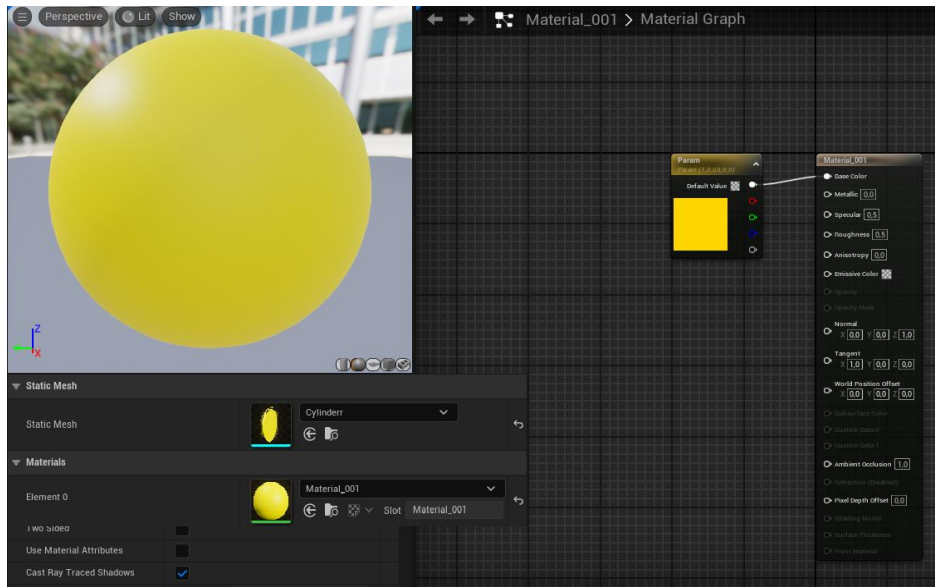
3.2.4.2.4 Kreiranje vlastitog 3D modela

Za gradski svijet kako bi igrači mogli zarađivati (osim dostavom pizze), kreiran je „Static Mesh“ novčića, objekt sa sferom oko sebe koji igrač sakupi i dodaje mu se u brojač novčića kada ga dotakne na ulici.



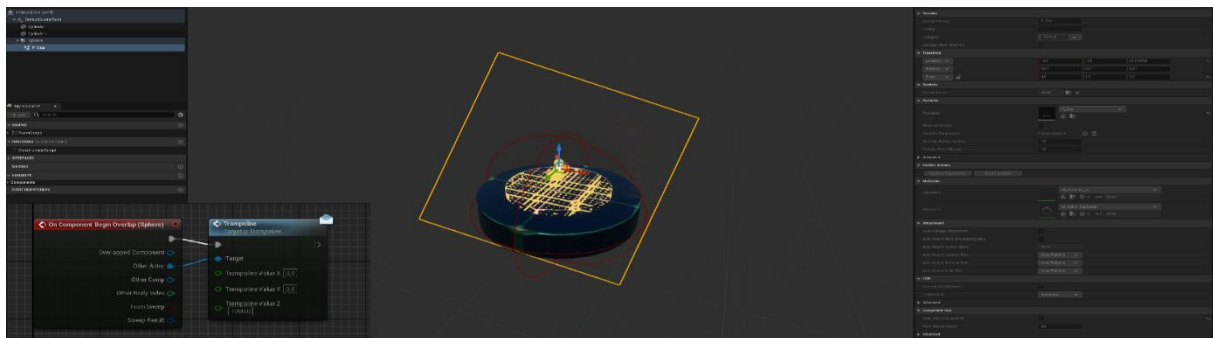
Slika 76 Kreiranje novčića

Ovdje je primjer dodavanja materijala novčiću kako bi zadobio krajnji izgled. Materijal je imitacija zlatne boje te samo kreiranje materijala za novčić prikazano je na slici ispod. Svaki „Mesh“ objekt ima svoj materijal i boje koje koristi.



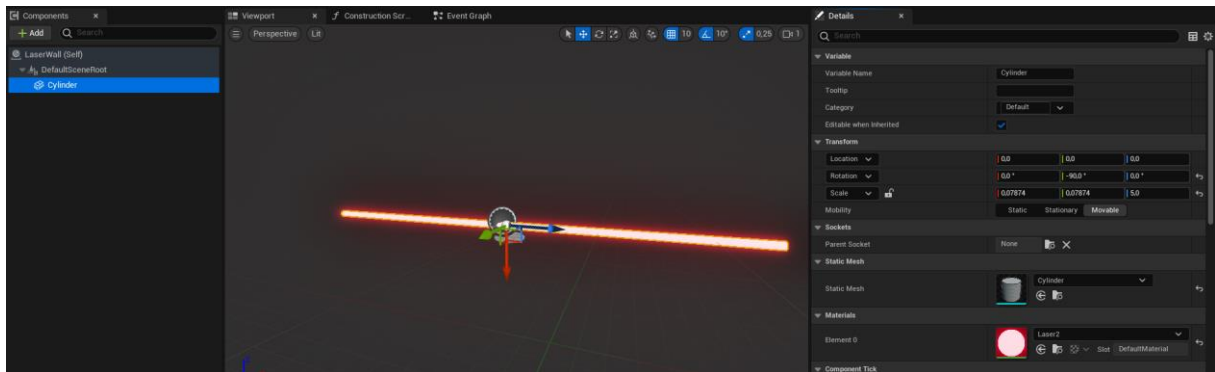
Slika 77 Materijal novčića

Za slučaj platformerskog svemirskog svijeta kreiran je trampolin kako bi igrač mogao odskočiti nekoliko metara u zrak te sletjeti sigurno na iduću platformu.

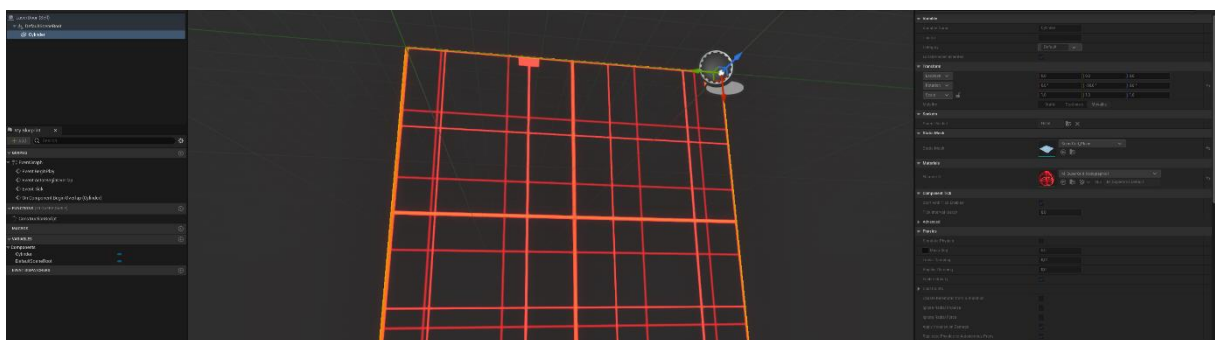


Slika 78 Kreiranje trampolina

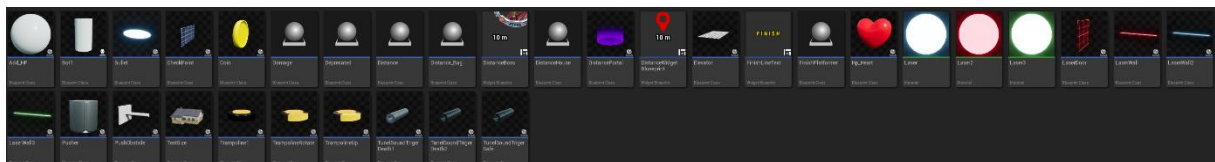
Također kreirani su zidni laseri kao i pokretna zidna ploča od cilindra holografskog stila.



Slika 79 Kreiranje zidnog hologramskog lasera



Slika 80 Kreiranje laserske hologramske ploče

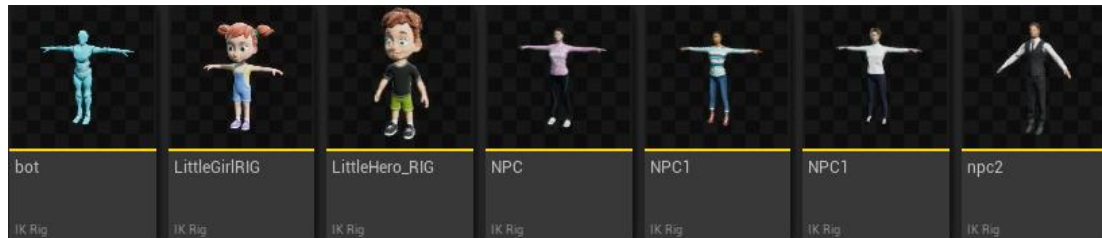


Slika 81 Ostali ručno kreirani funkcionalni objekti

3.2.4.2.5 Preusmjeravanje animacije (eng. Retargeter)

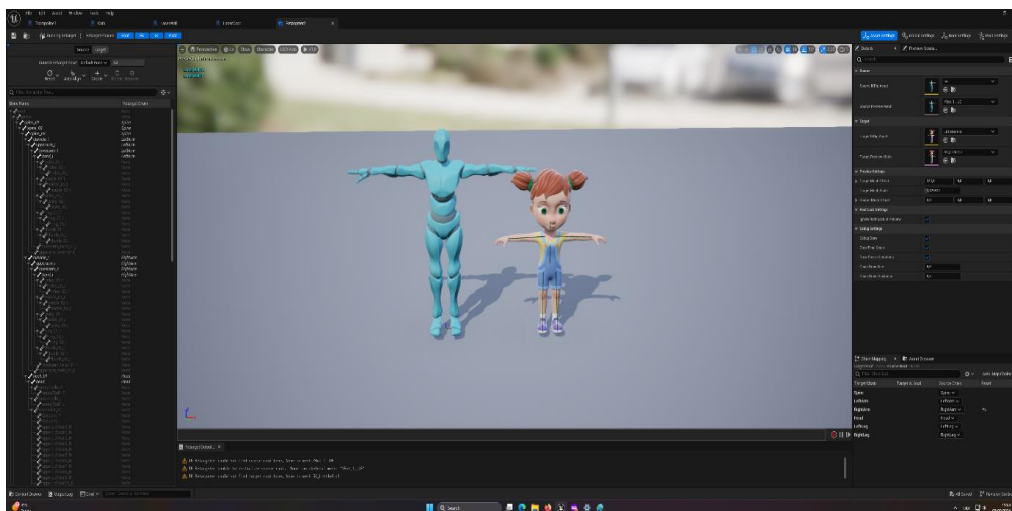
Kada se preuzmu animacije, u ovom slučaju sa stranice „Mixamo“, veliki broj animacija neće se ispravno pokretati na svakom kosturu.

Kako bi to izbjegli prvo se kreira „IK Rig Retargeter“ za željeni kostur te za kostur na kojem animacija uspješno radi.



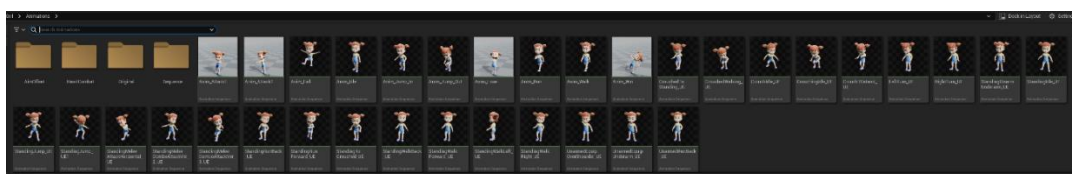
Slika 82 Primjeri "IK Rig" kostura

Zatim se prema slici ispod s lijeve strane nalazi kostur lika kojemu moramo postaviti željene točke glave, trupa i udova. Ako su točke identično postavljene za oba kostura, tada se one mogu spojiti, što je prikazano u desnom dolje kutu slike. Tada odabirom animacija po želji, animacija bi se trebala izvoditi ispravno na oba kostura.



Slika 83 Kosturi "IK Retargetera" testnog i ciljanog modela

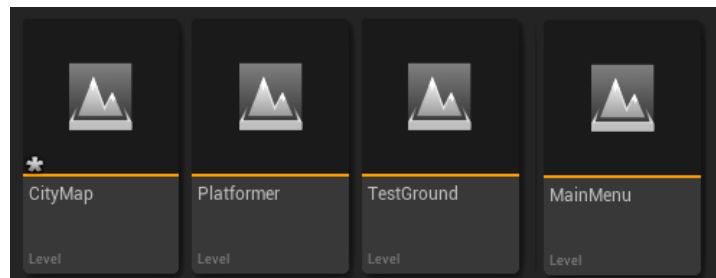
Ovaj proces vršimo na svim kosturima i animacijama identično.



Slika 84 Animacije igrača 2

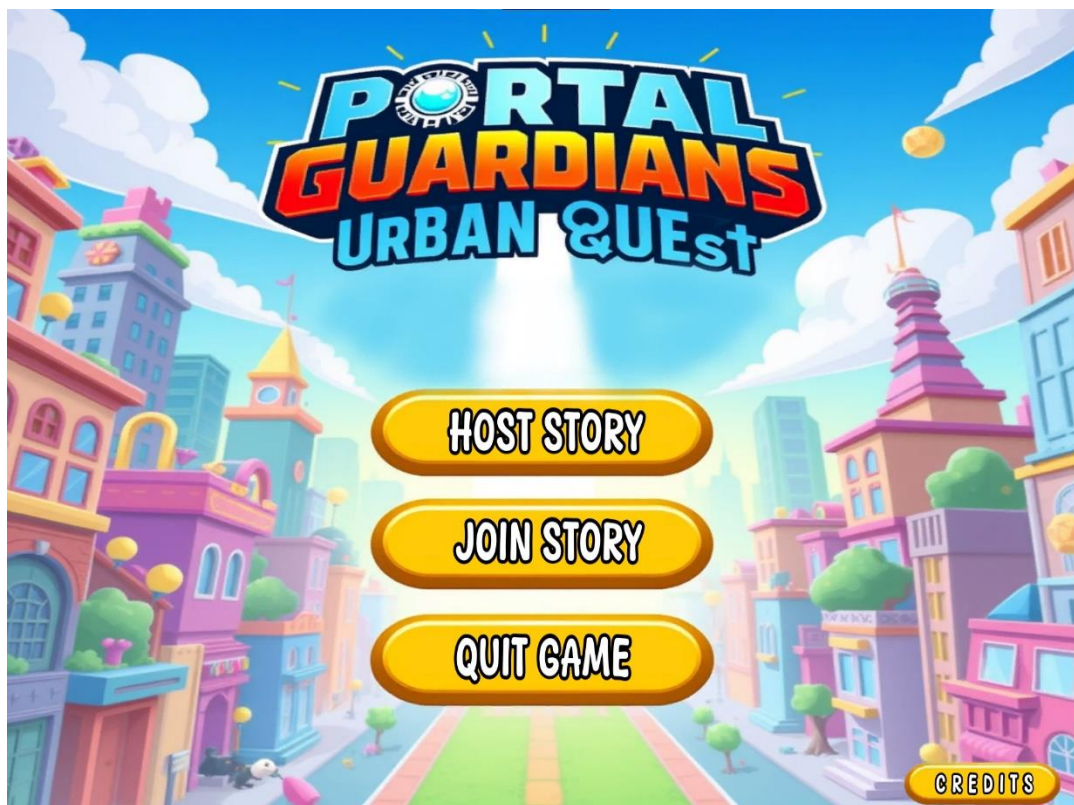
3.2.4.3 Razvijanje svjetova i dizajna okruženja igrača

Iako Epic Games u koaliciji s Unreal Engine-om nudi veliki broj gotovih i besplatnih 3D svjetova koje razvojni programeri mogu koristiti prilikom kreiranja videoigre, u ovom projektu kreirana su ručno dva svijeta pod nazivom „CityMap“ i „Platformer“. Uz njih na isti način kreiran je glavni izbornik koji je dizajniran uz pomoć korištenja umjetne inteligencije za generiranje fotografije izbornika te „TestGround“ svijet koji se koristio za pojedino testiranje funkcionalnosti sadržaja.



Slika 85 Klase svjetova

Prije ulaska u prvi svijet igra započinje u glavnom izborniku. Zbog jednostavnosti, implementirane opcije u izborniku su „Host Story“ i „Join Story“ koje omogućuju igraču 1 i igraču 2 da pokrenu svoju sesiju.



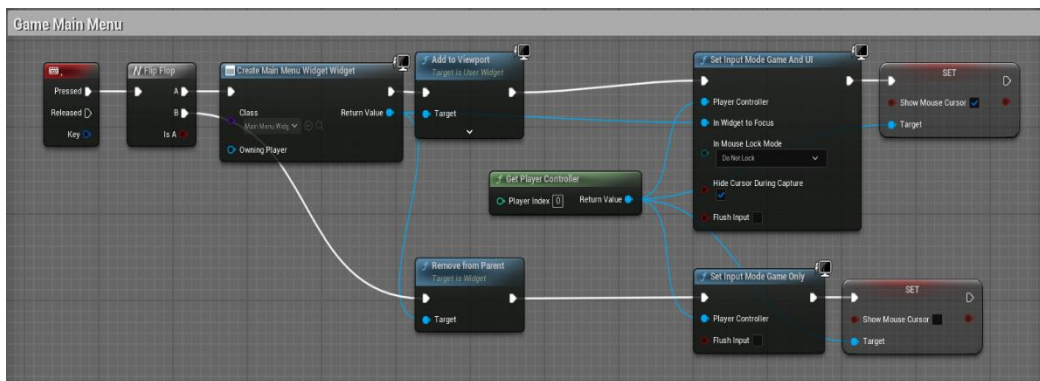
Slika 86 Glavni izbornik videoigre

Ono što treba napomenuti je da je videoigra razvijena na način da se pokreće za igrača 2 samo u slučaju kada igrač 1 pritisne „Host Story“, odnosno igrač 1 je domaćin sesije a igrač 2 je gost sesije. Sve dok domaćin ne pritisne gumb, igrač 2, odnosno gost će imati prozor učitavanja koji izgleda kao na slici 87.

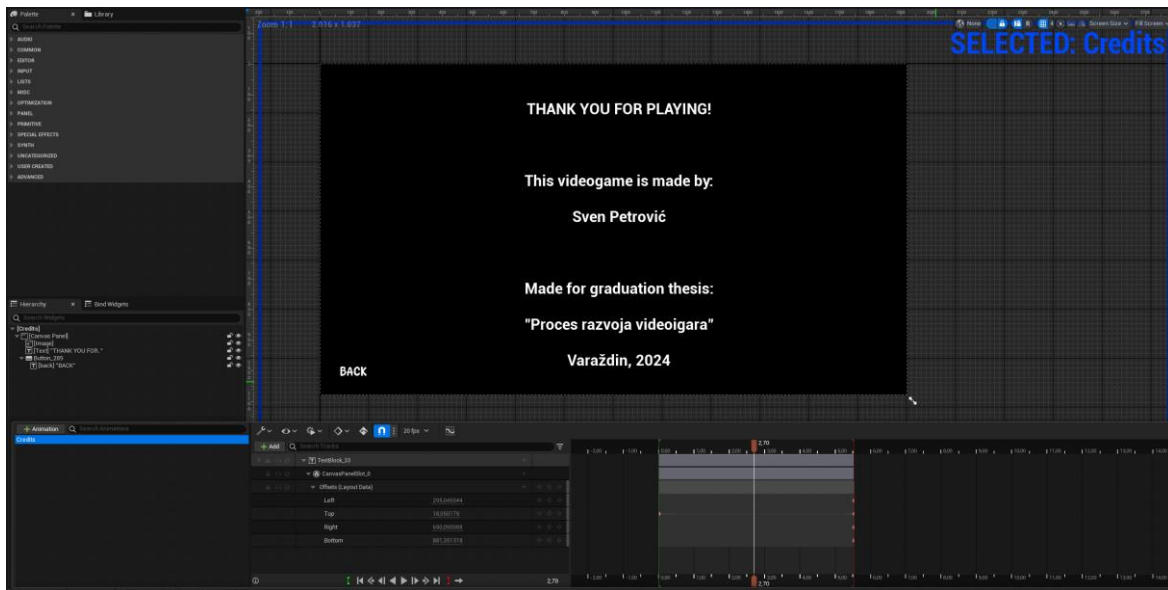


Slika 87 Prozor učitavanja sesije igrača 2

Pritiskom na određeni gumb na tipkovnici igračima je implementirana mogućnost odlaska u glavni izbornik te izlaz iz videoigre ili prikaz zasluge za videoigru (eng. credits).



Slika 88 Nacrtnog izbornika



Slika 89 Prikaz kreiranja zasluge videoigre

Prvi svijet dizajniran je u obliku jednog manjeg gradića sa svim osnovnim elementima koji jedan grad čini. Pod osnovne elemente misli se na ceste, kuće, automobile, garažu, restoran, dječje igralište i sl. Glavni dio priče videoigre kreće u ovome svijetu. To je realizirano na način da se definira start igrača (eng. Player Start), odnosno mjesto gdje će se igrači unutar svijeta stvoriti. U nastavku se nalaze slike najvažnijih lokacija ovog svijeta.



Slika 90 Svijet "CityMap"



Slika 91 Svijet "CityMap" - igralište



Slika 92 Svijet "CityMap" - kuća i portal starca

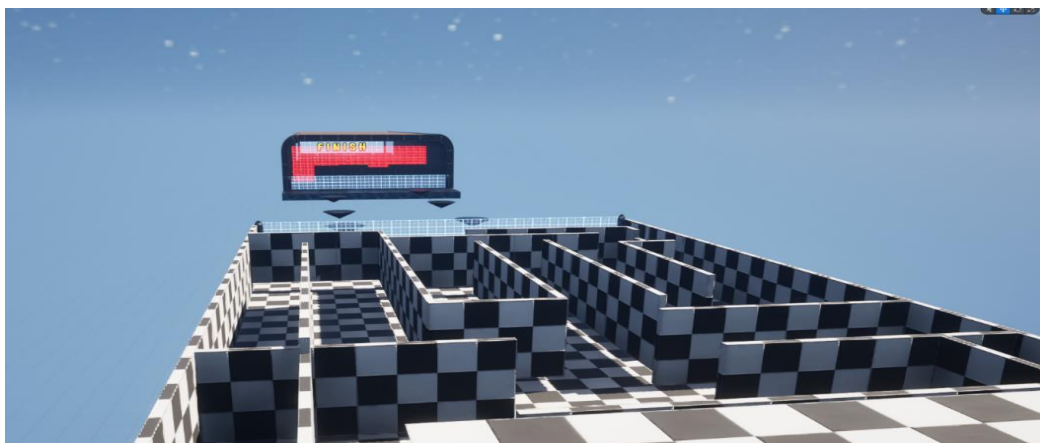


Slika 93 Svijet "CityMap" - garaža



Slika 94 Svijet "CityMap" - restoran

Drugi svijet (Platformer) dizajniran je s idejom da se igrači nalaze u izmišljenom tehnološkom svemiru te u scenariju ako ga prođu živi i zdravi spašavaju svijet i portal od zlih neprijateljskih robota. Zidovi i pod kreirani su ručno uz pomoć običnih geometrijskih likova koji se mogu naći u Unreal Engine-u. Ostali elementi poput stepenica i nestajućih blokova za preskakanje dodani su iz dodataka koje besplatno Epic Games trgovina nudi po nazivom „SuperGrid Starter pack“ [29].



Slika 95 Svijet "Platformer" - poligoni

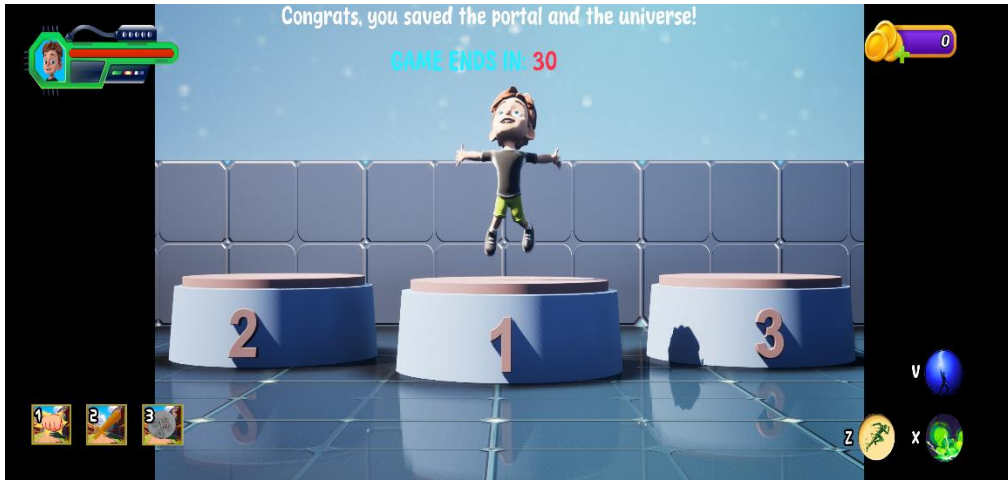
Laseri su kreirani također ručno, te su im dodane funkcionalnosti poput ozljeđivanja igrača.



Slika 96 Svijet "Platformer" - laseri

U ovom svijetu, za kretnju i nestajanje pojedinih tijela i objekata korištena je Engineova opcija kreiranja animacijskih sekvenci (iste one korištene za razradu priče videoigre) koje omogućuju kreiranje pokreta objekta po želji te čine videoigru težom. Svi ti objekti kreirani su u jednu cjelinu kako bi stvorili određene prepreke koje igrač mora proći kako bi bio nagrađen, odnosno došao do cilja i pobijedio. Prepreke se razlikuju, stoga daju određenu dozu zabave i natjecateljskog duha.

Kada igrači dođu do cilja, nalaze se na pobjedničkom postolju, ovisno koji su po redu došli do cilja.



Slika 97 Postolje

Nacr na slici ispod prikazuje kako se igrač teleportira na određenu lokaciju ulaskom u cilj, pod određenim kutom gledanja.



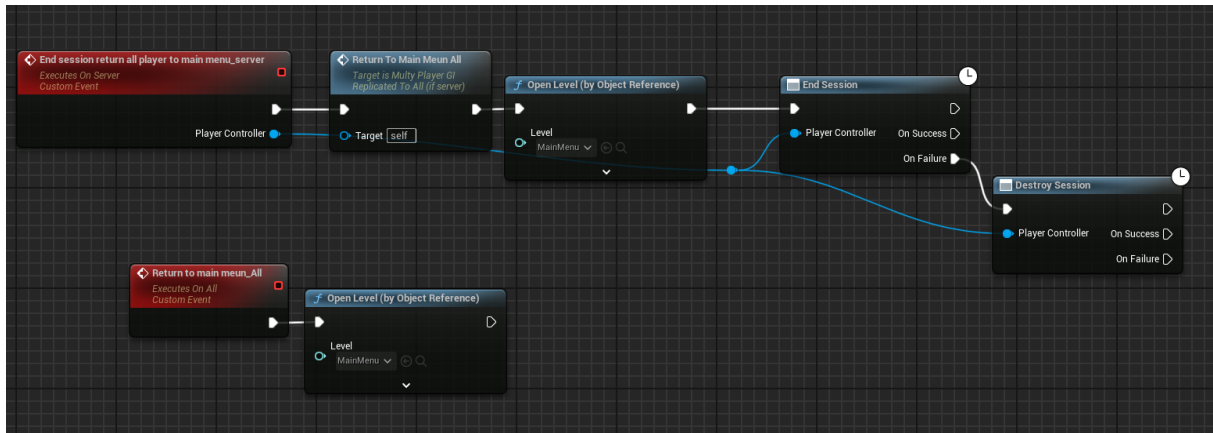
Slika 98 Nacr teleportacije ulaska u cilj

Teleportacija igrača na postolje ovisi o njihovom rednom broju ulaska u cilj. Ovisno o tome kreirali smo "Switch on Int" koji određuje poziciju sa svojim uvjetima izvršavanja.



Slika 99 Nacr teleportacije ulaska u cilj

Na kraju, prekida se sesija na način da se oba igrača prebace u glavni izbornik gdje mogu započeti sesiju iznova.



Slika 100 Kraj sesije videoigre

3.2.4.4 Integracija zvuka s vizualnim efektima

Proces integracije zvuka s vizualnim efektima unutar Unreal Engine-a poprilično je jednostavan proces, no ono što ovaj dio razvoja videoigre čini težim je pravilan odabir glazbe i zvučnih efekata koji daju jednu višu razinu doživljaja igračima.

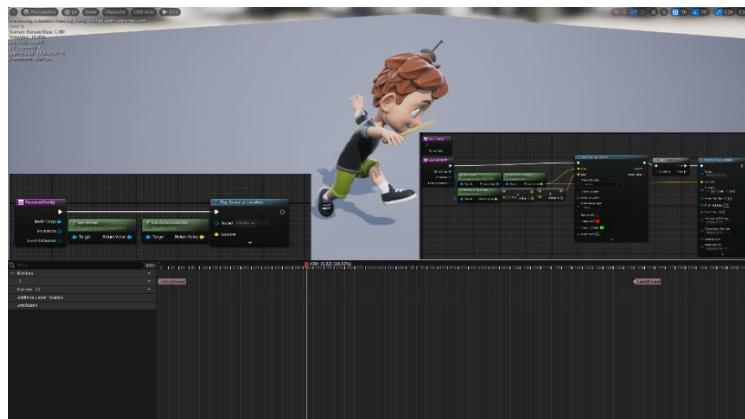
Kako se tema diplomskog rada drži određenih načela i procesa razvoja videoigre, tako se i u praktičnom dijelu držimo određene originalnosti (koliko je moguće) uporabe materijala za razvoj vlastite videoigre. Tako je za potrebu ovog diplomskog rada korištena glazba nepoznatog autora s našeg područja u svrhu promocije i otkrivanja još nepoznatih talenata. Za ostale efekte prilikom mehanika koristila se javna web platforma s besplatnim zvukovima i efektima za korištenje. [24]

Što se tiče simulacija zvukova ovisno o podlozi po kojoj glavni lik trči, dodajemo događaje „Animation Notify“.

Na fotografijama ispod, primjeri „Animation Notify“ dani su za animaciju lika za vrijeme trčanja („Footsteps“) te prilikom skoka („Jump sound“) i prilikom prizemljivanja („Land sound“). Ova funkcionalnost daje nam mogućnost odabira vremena javljanja pojedinog efekta zvuka kojeg možemo postaviti u određeno vrijeme animacije unutar trake animacije.

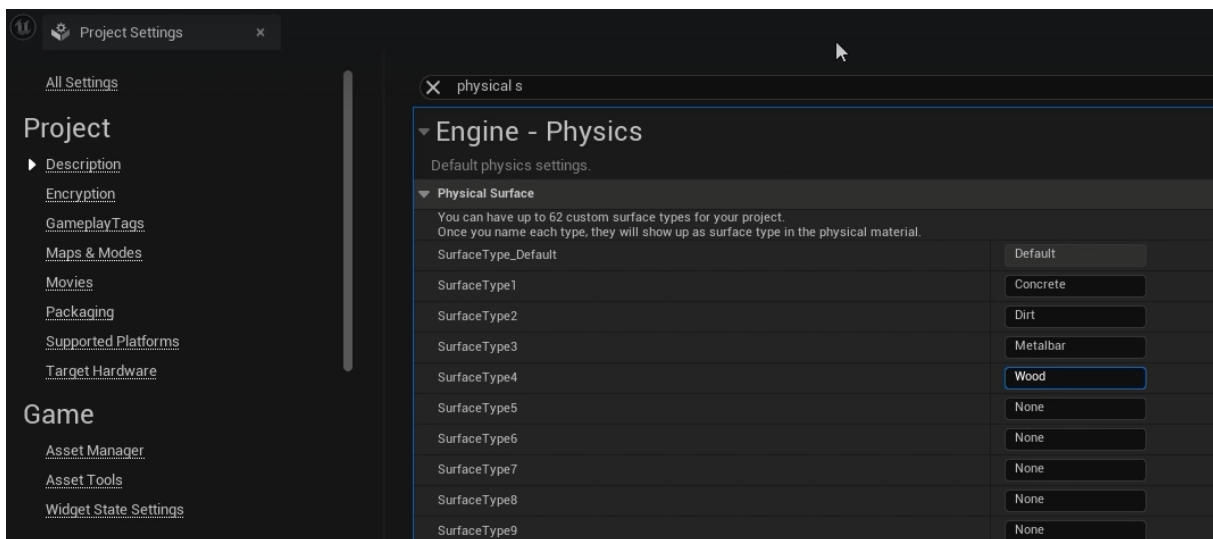


Slika 101 Animacija trčanja i njegova implementacija „Animation Notify“



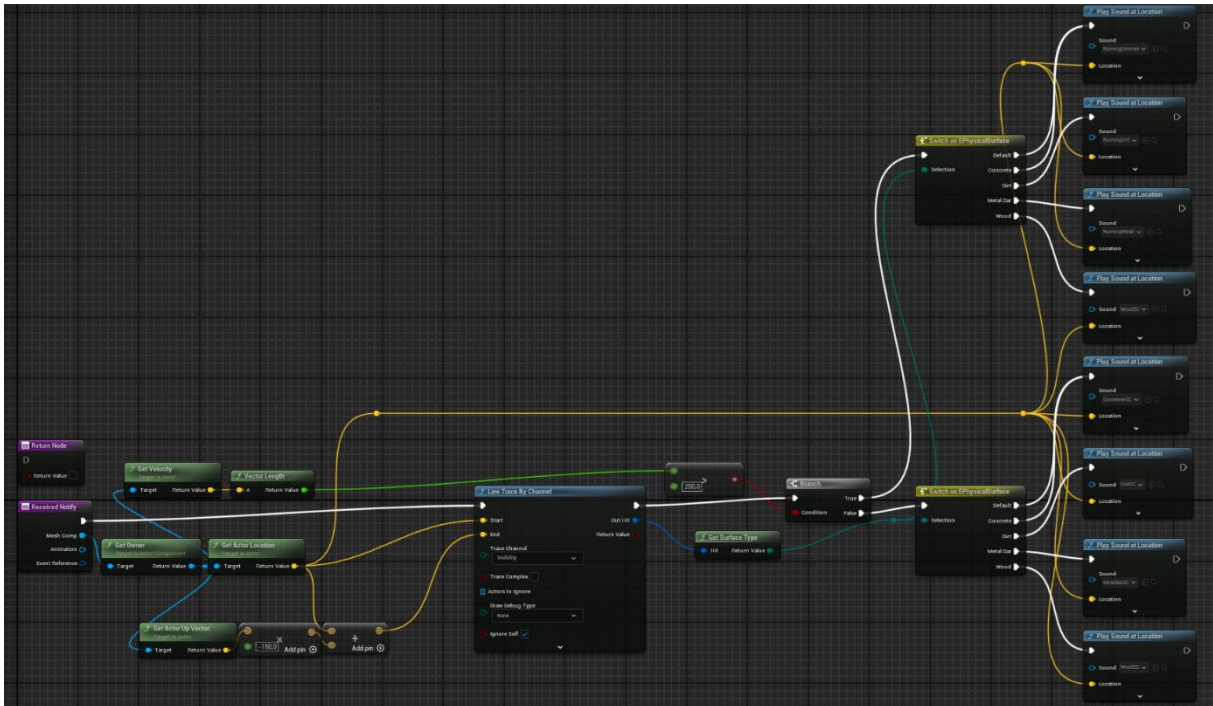
Slika 102 Animacija skoka lika i njegova implementacija nacrtu zvuka

Nadalje, kako bi softver prepoznao podlogu po kojoj igrač hoda, unutar postavki projekta morali smo definirati podloge poput betona, zemlje, drveta i metala što je prikazano na slici 103.



Slika 103 Postavke podloge

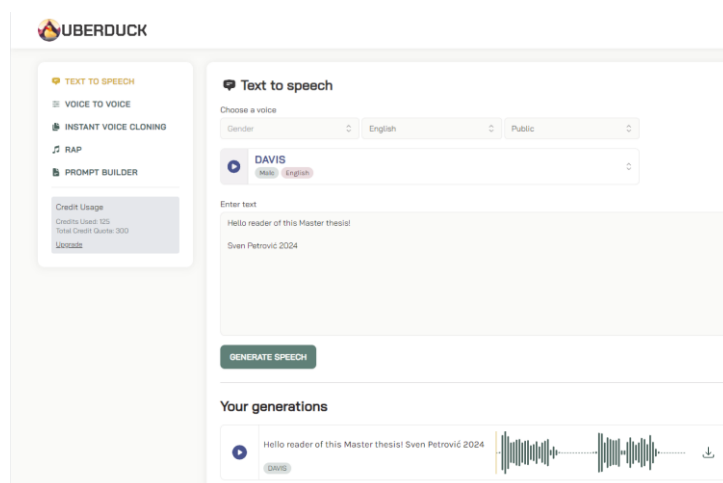
Nakon toga u nacrtu je bilo moguće koristiti „Switch on EPhysicalSurface“ čiji bi izlaz spojili s ulaznim zvukom ovisno o podlozi kao na slici 104.



Slika 104 Nacrt podloge

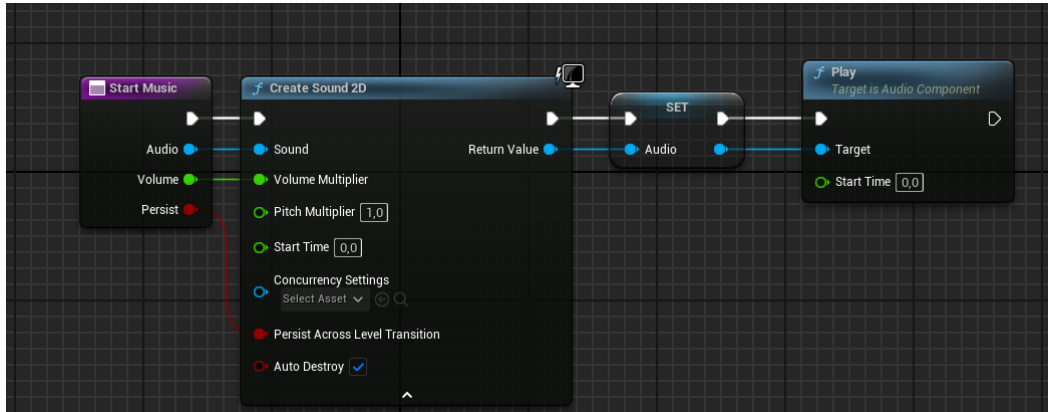
Uz ove primjere na identičan način uz pomoć „Play Sound at Location“ implementirani su zvukovi ozljeda i smrti glavnih likova, zvukove mača, lasera i sl.

Ono što također treba naglasiti je generiranje glasova pojedinog lika videoigre koja je realizirana putem javne web stranice „Uberduck“ koja putem umjetne inteligencije pretvara tekst u glasovni format ovisno o odabiru glasa po želji. [25] Generiranje se provodilo za dva glavna lika (djevojčicu i dječaka), starca, ženu s torbicom, konobara te neprijateljske robote.



Slika 105 Uberduck, glasovna umjetna inteligencija

Na samom kraju implementiran je zvuk koji se producira u glavnom izborniku, prilikom učitavanja svijeta na gostovoj sesiji te u svijetu „Platformer“. Sve što treba je u postavkama dodati naziv glazbe u formatu „WAV“ koju želimo izvoditi.



Slika 106 Nacrt implementacije glazbe za svjetove

3.2.5 Testiranje

3.2.5.1 Osiguranje kvalitete (eng. Quality Assurance Testing)

Prije nego što videoigra bude puštena na tržište, važno je, ali često zanemareno, osigurati da se videoigra ponaša prema predviđanjima, doseže očekivane standarde kvalitete krajnjih korisnika te ispunjava zahtjeve koje postavljaju proizvođači hardvera. To je trenutak kada na scenu stupa QA testiranje.

QA testiranje u videoigrama uključuje funkcionalno osiguranje kvalitete, koje osigurava tehničku spremnost igre za izdavanje, lokalizacijsko osiguranje kvalitete, koje ima za cilj pronaći i analizirati probleme u prijevodu igre te platformsku certifikaciju koja potvrđuje da su ispunjeni zahtjevi proizvođača hardvera unutar igre. Ovakva vrsta testiranja pomaže igri da prođe proces predaje (eng. submission), što je završni test koji određuje je li softver igre spreman za izdavanje. Prolazak ovog testa ključan je za dobivanje zelenog svjetla za prodaju igre, izbjegavanje kašnjenja u produkciji i izbjegavanje skupih naknada za ponovno podnošenje.[16]

3.2.5.2 Otklanjanje pogrešaka (Debugiranje)

Ova faza se bavi ispravljanjem grešaka u kodiranju, rješavanjem problema s kompatibilnošću i osiguravanjem da igra funkcionira onako kako je zamišljeno. Programeri identificiraju uzroke problema otkrivenih tijekom testiranja i brzo ih rješavaju. Ovaj proces traje iterativno sve dok igra ne postigne prihvatljivu razinu stabilnosti i očekivanje.

Neki od grešaka u praktičnom dijelu s kojima sam se susreo bili su od onih manjih kao što je npr. loša kutija udarca (eng. hitting box) lika koji nije stvarao štetu neprijateljima, stvaranje objekata na krivim mjestima ili različite veličine, određene video sekvence ne bi se pokretale te do nešto izazovnijih grešaka kao što je nevidljivost igrača 2 u prozoru igrača 1 ili spajanje nacrtu u krivi ulaz što bi katkada izazivalo grešku koju je unutar konzole teže pronaći.



Slika 107 Primjer greške prilikom stvaranja (eng. spawn) objekta na mapi

3.2.5.3 Testiranje performansi i optimizacija

Testiranje performansi je ključno za osiguranje da igra glatko radi na različitim platformama i uređajima. Tester procjenjuju elemente poput brzine kadrova, vremena učitavanja i ukupne responzivnosti. Napori optimizacije mogu uključivati prilagodbu koda, podešavanje grafičkih postavki ili pojednostavljenje grafičkih objekata u posebnom formatu (eng. assets) kako bi se poboljšale performanse igre. Ovaj korak je ključan za isporuku besprijekornog i ugodnog iskustva igrača.

Što se tiče praktičnog dijela diplomskog rada, zbog ograničenih resursa radne memorije, prije samog razvoja videoigre, softver ima razne mogućnosti optimiziranja performansi za efikasniji i brži rad nad samim razvojem videoigre. Tako su se primjerice za lakše svladavanje projekta koristile Engine-ove postavke kao što su:

- **Smanjenje skalabilnosti** - omogućava razvojnim programerima podešavanje razine detalja i vizualnih efekata unutar softvera. Za potrebu realizacije projekta, postavke su postavljene na „Low“ s „Epic“ konfiguracija. [18]
- **Nanite** - omogućuje renderiranje poligona u stvarnom vremenu bez žrtvovanja performansi. Ova značajka omogućuje kreatorima da dizajniraju vrlo detaljna okruženja i postave objekte bez brige o optimizaciji performansi na slabijem računalu. [18]
- **Minimalna upotreba dinamičkih sjena** - Dinamičke sjene mogu poboljšati realizam u igrama, no u slučaju da su vam potrebne dinamičke sjene, samo pokušajte ograničiti broj dinamičkih svjetala, a zatim prilagodite razlučivost i udaljenost sjene kako biste balansirali između kvalitete i performansi. Smanjenjem dinamičkog renderiranja sjena i fokusiranjem na statičko osvjetljenje, performanse na slabijim sustavima mogu se znatno poboljšati. [18]
- **Smanjiti broj osvjetljenja u okruženju videoigre** - dinamička svjetla također mogu negativno utjecati na performanse jer su zahtjevna za resurse. Stoga ograničite broj dinamičkih svjetala i koristite statička gdje god je to moguće. [18]

Iz ovog razloga testiranje performansi znatno je otežano te je nešto što svakako treba uzeti u obzir kod razvijanja nešto većeg projekta. Konkretnu razinu kvalitete performansi u ovom slučaju bile su moguće tek nakon završetka videoigre, pokretanjem na jačem računalu. U slučaju daljnjeg razvoja, kvaliteta performansi, uključujući osvjetljenje i sjene bile bi jedan od glavnih ciljeva u budućem razvoju.

3.2.5.4 Dokumentacija i izvještavanje

Kao što je spomenuto prije, tijekom testiranja održava se detaljna dokumentacija. Testerima dokumentiraju korake koji su poduzeti, identificirane probleme i rezultate testiranja. Izvještaji se također generiraju i dijele s razvojnim timom, pružajući jasne uvide u status igre i napredak u rješavanju problema putem identifikacije svakog pojedinačno.

U nastavku je prikazana tablica s primjerima dokumentiranih funkcionalnosti i njihovih pogrešaka za lakše razumijevanje provođenja dokumentacije i izvještaja prilikom razvijanja i testiranja:

Tablica 2 Primjer dokumentiranih funkcionalnosti i njihovih pogrešaka (Izvor:[16])

Vrsta problema/pogreške	Naziv problema/pogreške	Primjer problema/pogreške	Posljedice problema/pogreške	Uspješnost procesa
Performanse	Zamrzivanje	Videoigra se zamrzava x puta	Nespremljeni podaci će biti izgubljeni.	Neuspješan
Kamera	Krivi kut gledanja	U prvom licu pucačka igra prikazuje se u trećem licu	Mogućnost zbunjivanja igrača. Nemogućnost napretka.	Neuspješan
Grafički	Nepotpunost u realizaciji grafike	Netočno implementirana tekstura	Nepotpun doživljaj videoigre	Postoji mogućnost uspjeha
Kolizija	Nema kolizije	Igrač može prolaziti kroz zidove ili pasti s mape	Nemogućnost napretka u videoigri	Moguć uspjeh
Umjetna inteligencija	NPC je zapeo u petlji	NPC vrši iste radnje više puta	Videoigra izgleda nedovršeno	Moguć uspjeh
Audio	Vrijeme izvršavanja	Audio se reproducira u krivo vrijeme.	Korisnik bi mogao biti zbunjen i ne razumjeti zvuk i razvoj događaja.	Moguć uspjeh

3.2.5.5 Beta i ostale verzije testiranja

Beta testiranje uključuje objavljivanje pred finalne verzije igre odabranoj skupini vanjskih korisnika/igrača ili demo verzije. Ovi korisnici, često predstavljaju ciljanju publiku videoigre, pružaju vrijedne povratne informacije iz različitih perspektiva. Beta testiranje može otkriti probleme koji nisu identificirani tijekom unutarnjeg QA testiranja. Također služi kao prilika za prikupljanje uvida u preferencije i očekivanja igrača koje se mogu ažurirati u budućnosti.

Postoje tri faze testiranja :

- **Faza zatvorene alfe** - strogo kontrolirana i uključuje osnovnu verziju glavnih likova, predstavljenu jednostavnim skinovima koji se mogu kretati po ekranu i sudarati s osnovnim oblicima platformi. Testeri su često pažljivo odabrani unutarnji zaposlenici, developeri i stručnjaci iz industrije. Ova faza ima za cilj identificirati kritične probleme, poput grešaka (eng. bugs) koji mogu spriječiti igranje igre. [17]
- **Faza zatvorene bete** - developeri testiraju sve razine videoigre s ažuriranim kostimima likova i sposobnostima kretanja s većom grupom beta testera. Stotine ili čak tisuće odabranih igrača pružaju povratne informacije o mehanici igre, tempu i iskustvu, pomažući developerima da usavrše razinu težine, sustave napredovanja i nagrade. [17]
- **Faza otvorene bete** - igra je gotovo završena. Sve razine su gotove te nema značajnih novih likova ili neprijatelja koji se uvode. Svi problemi iz prethodnih faza moraju biti riješeni prije javne bete. Javna beta omogućuje svima zainteresiranima da sudjeluju. Ovo testiranje na velikoj skali pomaže otkriti mnoge probleme, greške i rizične slučajeve koji su možda promašeni u prethodnim fazama. [17]

3.2.6 Faza prije lansiranja

3.2.6.1 Marketing

Faza prije lansiranja sadržaja također je obilježena intenzivnim marketinškim i promotivnim aktivnostima kako bi se stvorila uzbuđenja i svijest o nadolazećoj igri. To uključuje objavljivanje trailer-a, teaser-a i drugih promotivnih materijala. Razvojni timovi mogu se angažirati s influencerima, novinarima iz industrije igara i kreatorima sadržaja kako bi generirali uzbuđenje oko igre. Kampanje na društvenim mrežama, priopćenja za tisak i druge marketinške strategije koriste se za doseganje ciljne publike. [9]

3.2.6.2 Postavljanje sustava podrške

U iščekivanju upita i problema igrača, programeri postavljaju sustave podrške, uključujući kanale za korisničku službu, FAQ i forume zajednice. Cilj je osigurati da igrači imaju pristup pomoći i informacijama, pozitivno iskustvo i rješenje za bilo kakav problem koji može nastati tijekom početne faze lansiranja.

U takvom angažmanu sa zajednicom bitno je da razvojni timovi pružaju povratne informacije, sudjeluju na forumima i odgovaraju na upite igrača. Izgradnja pozitivnih odnosa sa zajednicom doprinosi dugoročnom zadržavanju igrača i može pružiti vrijedne uvide za buduće nadogradnje i poboljšanja.

3.2.7 Lansiranje

3.2.7.1 Distribucija videoigre, praćenje i rješavanje problema

Proces lansiranja započinje distribucijom igre na različite platforme. To može uključivati objavljivanje naslova na digitalnim distribucijskim platformama poput STEAM-a, Epic Games Store-a, trgovina aplikacijama za mobilne igre ili konzolnih platformi. Razvojni tim osigurava da su datoteke igre ispravno prenesene i da su svi potrebni podaci, poput cijene i sistemskih zahtjeva točni.

Pazi se na lansiranje kako bi se riješili svi neposredni problemi koji se mogu pojaviti. To uključuje praćenje opterećenja servera, rješavanje potencijalnih tehničkih kvarova i osiguranje da igrači mogu pristupiti i preuzeti igru bez značajnih problema. Alati za praćenje u stvarnom vremenu mogu se koristiti za prikupljanje podataka o aktivnosti igrača, performansama servera i drugim ključnim metrima.

Odmah nakon lansiranja mogu se pojaviti neočekivani problemi. Timovi su na raspolaganju za rješavanje neposrednih problema, kao što su kvarovi servera, problemi s povezivanjem ili nepredviđeni tehnički problemi. Kanali komunikacije, uključujući društvene

mreže, forume zajednice i korisničku podršku, aktivno se prate kako bi se pružili brzi odgovori na upite igrača ili izvještaje o problemima.

3.2.8 Faza nakon lansiranja

3.2.8.1 Praćenje povratnih informacija igrača i ispravak grešaka nakon lansiranja

Kontinuirano praćenje povratnih informacija igrača ključan je aspekt faze koja slijedi nakon produkcije. Aktivno slušanje zajednice, uključujući forume, društvene mreže i druge komunikacijske kanale, pospješuje razumijevanje iskustva i brige igrača.

Analiziranje povratnih informacija pomaže identificirati područja koja zahtijevaju pažnju i promjene, bilo da se radi o ispravljanju grešaka, poboljšanju elemenata igre ili implementaciji značajki koje su tražili igrači. Razvojni tim prioritizira identificirane probleme na temelju ozbiljnosti i utjecaja na iskustvo igrača.

3.2.8.2 Planiranje, razvoj i nadogradnja dodatnog sadržaja

Redovite nadogradnje objavljuju se s ciljem da se uvedu nove značajke, poboljšanja i popravci. Ove nadogradnje mogu uključivati optimizacije performansi i prilagodbe ravnoteže. Raspored objavljivanja nadogradnji često proizlazi iz povratnih informacija od igrača i ovisi o unutarnjim prioritetima razvojnog tima.

Pružanje redovitih softverskih ažuriranja za videoigru mogu također uključivati zakrpe za balansiranje igre ili novi sadržaj za preuzimanje (DLC). U današnjoj industriji videoigara uvođenje novog sadržaja uobičajeno je jer povećava vrijednost ponovnog igranja i privlačnost igre. Novi nivoi, priče i multiplayer modovi samo su neke od opcija DLC-a koje razvojni studio može istražiti. [19]

Ono što jedan DLC može sa sobom donijeti su osvježene nove mape i razine, novi likovi i njihova odjeća (eng. skins), razni novi modovi i novi neprijatelji. U novije vrijeme postoji i pretplata na premiju videoigre koja traje nekoliko tjedana gdje igrači mogu osvajati posebne nagrade. Ovakvim unaprijed planiranjem i nadogradnjom pospješuje se održavanje koncentracije igrača.

3.2.9 Lansiranje videoigre na tržište

3.2.9.1 Platforma Steam

Prije samog pojašnjenja procesa lansiranja videoigre, potrebno je napomenuti kako proces lansiranja za ovu videoigru nije realiziran u potpunosti ponajviše zbog korištenja velikog broja 3D modela koji su dostupni svima, odnosno zbog manjka originalnih 3D modela i zbog toga što je videoigra u ranoj beta verziji, ali s dobrom podlogom za napredak. Kako se radi o videoigri koju podržava PC, primjer lansiranja proizvoda na tržište opisat ćemo za jednu od glavnih platformi za lansiranje svoje nove videoigre, a to je Steam.

Steam je jedna od najvećih platformi za digitalnu distribuciju videoigara, koju je razvila kompanija Valve i lansirala 2003. godine. U početku je služila kao alat za automatsko ažuriranje igara, ali se ubrzo proširila u najveće tržište za kupovinu i distribuciju videoigara. Do 2021. godine, Steam je imao 120 milijuna aktivnih korisnika mjesečno i katalog od preko 50 tisuća igara. Platforma je postala ključna za razvojne studije, uključujući i neovisne "indie" developere, omogućujući im širok pristup globalnom tržištu. Steam također igra važnu ulogu u prikupljanju korisničkih recenzija, koje mogu biti ključne za poboljšanje igara i donošenje poslovnih odluka. [20]

Platforma pruža ključnu podršku za multiplayer igre, omogućujući igračima globalno povezivanje i zajedničko igranje u realnom vremenu. Platforma podržava lokalne i online „multiplayer modove“, kao i „cross-platform“ igranje. Kroz alate poput Steamworks API-a i podrške za „dedicated“ servere, developerima je olakšana integracija multiplayer funkcionalnosti u igre. [20]

Ovakva podrška čini Steam preferiranom platformom za distribuciju i igranje „multiplayer“ videoigara, stoga je i odličan kandidat za demonstraciju. U idućem ulomku slijedi primjer, pravila i koraci kojih se svaki developer treba pridržavati prije prvog lansiranja videoigre.

3.2.9.2 Pravila i koraci lansiranja

Kako Steam nalaže na svojoj javnoj web stranici, prvi korak procesa uključuje potpisivanje digitalne dokumentacije, što zahtijeva unos bankovnih i poreznih informacija, kao i potvrdu identiteta. Nakon toga, potrebno je platiti naknadu za distribuciju aplikacije u iznosu od 100 američkih dolara putem podržanih Steam metoda plaćanja.

Nakon ispunjavanja dokumentacije, dobivate pristup Steamworks alatima koji vam omogućuju pripremu vaše igre za lansiranje. To uključuje izradu stranice trgovine, prijenos verzija igre, konfiguraciju značajki te postavljanje željene cijene. Prije nego što vaša videoigra postane dostupna, prolazi kratki proces pregleda, gdje Steam provjerava ispravnost i sigurnost vašeg proizvoda. Ovaj proces obično traje između 1 i 5 dana.

Postoje i određeni vremenski zahtjevi prije nego što možete lansirati svoju igru. Naime, mora proći 30 dana od trenutka plaćanja naknade do mogućnosti objavljivanja igre, što Steam-u omogućuje provjeru vaših podataka. Prilikom lansiranja igre, važno je pridržavati se Steamovih pravila i smjernica vezanih za sadržaj:

- Zabranjeno je objavljivati govor mržnje
- Zabranjeno je objavljivati seksualno eksplicitni sadržaj
- Zabranjene su klevete i slične izjave
- Zabranjen je sadržaj koji krši zakone
- Zabranjeno je objavljivati zloćudne softvere i slične sadržaje

Steam primarno prihvaća igre za distribuciju, ali može prihvatiti i druge vrste softvera ako spadaju u određene kategorije kao što su animacija, audio/video produkcija, ili alati za uređivanje.

3.2.9.3 Mrežni podsustav STEAM-a i API

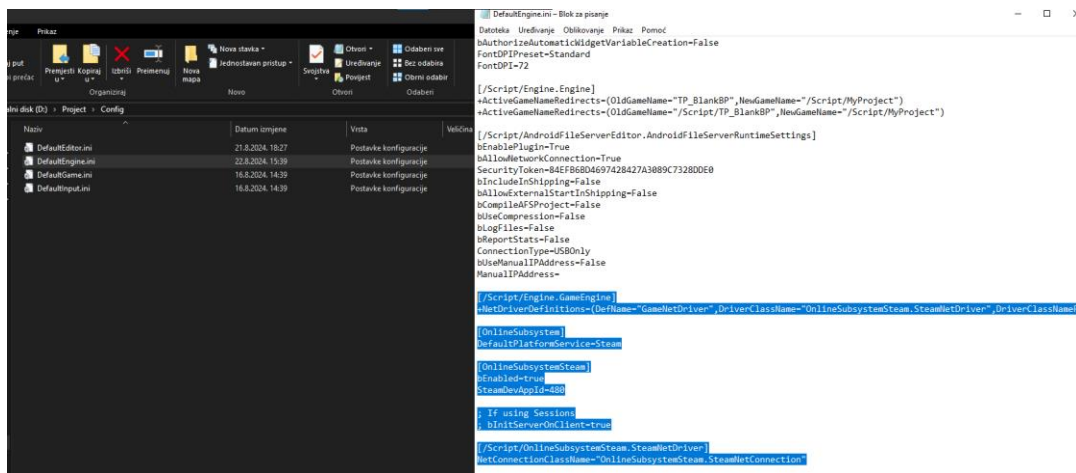
Za potrebe ovog projekta, koristio se STEAM-ov aktivni mrežni (eng. Online) podsustav dostupan svima u svrhu spajanja vlastite videoigre namijenjene za više igrača sa STEAM platformom.

„Online Subsystem Steam API“ omogućuje vam distribuciju aplikacija izrađenih u Unreal Engine-u na Valveovu Steam platformu. Glavna svrha Steam modula je pružiti set funkcionalnosti. Osim toga, Steam modul implementira nekoliko sučelja koja su dostupna putem „Online Subsystem“-a, podržavajući većinu onoga što nudi „Steamworks Software Development Kit“ (SDK).

Dostupna Steam sučelja, odnosno funkcionalnosti uključuju:

- Usklađivanje igrača (*eng. Lobbies*)
- Ljestvice (*eng. Leaderboards*)
- Postignuća (*eng. Achievements*)
- Glasovni chat (*eng. Voice chat*)
- Korisnički oblak (*eng. User Cloud*)
- Dijeljeni oblak (*eng. Shared Cloud*)
- Vanjsko korisničko sučelje (*eng. External UI*)

Kako bi ovo realizirali, potrebno je provesti korak implementacije dodatnih postavki u svoj projekt na način da iduće komande dodamo u svoju datoteku pod nazivom "DefaultEngine.ini" koja se nalazi u "Config" mapi projekta u Unreal Engine-u [22]:



Slika 108 Implementacija komandi za STEAM platformu

Idući korak je dodavanje podataka koji su dostupni na Github platformi [23] i javni su za preuzimanje a tiču se sesija koje su u interakciji sa STEAM-om. Dva dodatka koja smo dodali za slučaj praktičnog dijela rada su „AdvancedSessions“ i „AdvanceddSteamSessions“.

Prije pokretanja projekta, potrebno je ručno „buildanje“ projektne datoteke formata „.sln“ za koju je ključan razvojni program Visual Studio 2022 kako bi se ove promjene postavki primijenile. Ukoliko se ne odlučite na ručni „build“ Unreal Engine u većini slučajeva prepozna promjene te će to učiniti umjesto vas ako se verzije softvera poklapaju.

Posljednji korak je da se ulogirate u svoj Steam račun, pokrenete projekt i testirate prepoznaje li Steam videoigru aktivnom na način da se u desnom dolje kutu prikaže aktivnost kao na slici ispod.



Slika 109 Steam - prepoznavanje videoigre

4. Zaključak

Prilikom razvoja vlastite videoigre uz pomoć programskog alata Unreal Engine 5, došao sam do zaključka kako su faze procesa razvoja videoigre u teoriji vrlo slične onima u stvarnome životu. Nova i napredna tehnologija, koju današnji razvojni programi i alati pružaju, značajno olakšava realizaciju ovakvog projekta za pojedinca.

Praktični dio diplomskog rada uvelike je doprinio dobivanju jasne slike o vremenskom utrošku i kompleksnosti svake faze razvoja videoigre, od samog kreiranja ideje i priče, preko odabira odgovarajućih 3D modela i animacija, pa sve do analize i učenja mehanika implementacije, testiranja projekta i same produkcije.

Proces razvoja omogućio je dublje razumijevanje ključnih koraka u razvoju videoigre, poput balansiranja tehničkih izazova s kreativnim vizijama. Kroz ovaj projekt potvrdilo se da su strpljenje, detaljno planiranje, ispravna implementacija i konstantno testiranje ključni faktori za uspješan razvoj igre. Uz to, nova verzija Unreal Engine-a omogućila je lakšu primjenu naprednih značajki, čime je finalni proizvod tehnički usavršen i prilagođen zahtjevima današnje industrije videoigara.

Ovaj diplomski rad potvrdio je važnost svakog pojedinog ciklusa u razvoju videoigre te pružio praktično iskustvo koje je obogatilo moje razumijevanje ovog dinamičnog i kreativnog procesa čime sam stekao i nova znanja.

Osim tehničkog i kreativnog aspekta razvoja, važan dio svakog projekta je i plasiranje gotovog proizvoda na tržište. Prilikom lansiranja videoigre ključno je razmotriti strategije distribucije, marketinga i promocije kako bi igra dosegla što širu publiku. Digitalne platforme poput Steama, Epic Games Storea i sličnih pružaju izvrsne prilike za distribuciju igara neovisnim razvojnim programerima te su lako dostupne svima.

5. Popis literature

- [1] „What are Video Games? - Fun Facts about Video Games for Students and Teachers“. Pristupljeno: 29. kolovoz 2024. [Na internetu]. Dostupno na: <https://www.twinkl.hr/teaching-wiki/video-games>
- [2] I. Granic i A. Lobel, „The Benefits of Playing Video Games“, American Psychologist, 2014.
- [3] „Unreal Engine 5 features that will change the gaming industry“. Pristupljeno: 29. kolovoz 2024. [Na internetu]. Dostupno na: <https://mainleaf.com/unreal-engine-5-features/>
- [4] „Mixing blueprints and C++ | Community tutorial“, Epic Games Developer. Pristupljeno: 29. kolovoz 2024. [Na internetu]. Dostupno na: <https://dev.epicgames.com/community/learning/tutorials/YDpo/mixing-blueprints-and-c>
- [5] B. Foundation, „About“, blender.org. Pristupljeno: 29. kolovoz 2024. [Na internetu]. Dostupno na: <https://www.blender.org/about/>
- [6] „Online Courses - Learn Anything, On Your Schedule“, Udemy. Pristupljeno: 29. kolovoz 2024. [Na internetu]. Dostupno na: <https://www.udemy.com/join/passwordless-auth/>
- [7] „Game Development Stages: How are video games created?“, Walla Walla Studio. Pristupljeno: 29. kolovoz 2024. [Na internetu]. Dostupno na: <https://wallawallastudio.com/article/game-development-stages/>
- [8] Y. Denisyuk, „What are the stages of game development?“, Pingle Studio. Pristupljeno: 29. kolovoz 2024. [Na internetu]. Dostupno na: <https://pinglestudio.com/blog/full-cycle-development/game-development-stages>
- [9] „7 stages of game development | Planning stage“. Pristupljeno: 29. kolovoz 2024. [Na internetu]. Dostupno na: <https://stepico.com/blog/game-development-stages/>
- [10] T. Solod, „What’s the Most Popular Gaming Platform in 2023?“, Pingle Studio. Pristupljeno: 29. kolovoz 2024. [Na internetu]. Dostupno na: <https://pinglestudio.com/blog/industry-news/whats-the-most-popular-gaming-platform-in-2023>
- [11] P. Hollstrand, Supporting Pre-Production in Game Development : Process Mapping and Principles for a Procedural Prototyping Tool. 2020. Pristupljeno: 30. kolovoz 2024. [Na internetu]. Dostupno na: <https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-273926>
- [12] K. Sulopuisto, „LEAN PRE-PRODUCTION FOR INDEPENDENT GAME DEVELOPMENT“

- [13] „Hardware and Software Specifications for Unreal Engine | Unreal Engine 5.4 Documentation | Epic Developer Community“, Epic Games Developer. Pristupljeno: 30. kolovoz 2024. [Na internetu]. Dostupno na: <https://dev.epicgames.com/documentation/en-us/unreal-engine/hardware-and-software-specifications-for-unreal-engine>
- [14] „Blender on Steam“. Pristupljeno: 30. kolovoz 2024. [Na internetu]. Dostupno na: <https://store.steampowered.com/app/365670/Blender/>
- [15] H. M. Chandler, The Game Production Handbook. Jones & Bartlett Publishers, 2009.
- [16] E. Ruuska, „QUALITY ASSURANCE TESTING IN VIDEO GAMES“.
- [17] Testlio, „Beta Testing for Video Games: What It Is and Why You Need It“, Testlio. Pristupljeno: 31. kolovoz 2024. [Na internetu]. Dostupno na: <https://testlio.com/blog/crowdsourced-beta-testing/>
- [18] H. Trieu, „Optimize Unreal Engine 5 performance on low-end computers | iRender“, iRender Cloud Rendering Service. Pristupljeno: 31. kolovoz 2024. [Na internetu]. Dostupno na: <https://irendering.net/optimize-unreal-engine-5-performance-on-low-end-computers/>
- [19] „From Idea to Launch: The 7 Stages of Game Development“, G2. Pristupljeno: 31. kolovoz 2024. [Na internetu]. Dostupno na: <https://www.g2.com/articles/stages-of-game-development>
- [20] M. N. Rizani, M. N. A. Khalid, i H. Iida, „Application of Meta-Gaming Concept to the Publishing Platform: Analysis of the Steam Games Platform“, Information, sv. 14, izd. 2, Art. izd. 2, velj. 2023, doi: 10.3390/info14020110
- [21] „Steamworks Partner Program“. Pristupljeno: 01. rujan 2024. [Na internetu]. Dostupno na: <https://partner.steamgames.com/steamdirect>
- [22] „Online Subsystem Steam | Unreal Engine 4.27 Documentation | Epic Developer Community“, Epic Games Developer. Pristupljeno: 02. rujan 2024. [Na internetu]. Dostupno na: <https://dev.epicgames.com/documentation/en-us/unreal-engine/online-subsystem-steam-interface-in-unreal-engine>
- [23] Risensy, Risensy/Steam. (29. kolovoz 2024.). C++. Pristupljeno: 02. rujan 2024. [Na internetu]. Dostupno na: <https://github.com/Risensy/Steam>
- [24] „90,000+ Free Sound Effects for Download - Pixabay“. Pristupljeno: 02. rujan 2024. [Na internetu]. Dostupno na: <https://pixabay.com/sound-effects/>

[25] „AI Vocals and Text To Speech | Uberduck“. Pristupljeno: 02. rujan 2024. [Na internetu]. Dostupno na: <https://www.uberduck.ai>

[26] „Little Hero in Characters - UE Marketplace“, Unreal Engine. Pristupljeno: 02. rujan 2024. [Na internetu]. Dostupno na: <https://www.unrealengine.com/marketplace/en-US/product/little-hero>

[27] „Little Girl in Characters - UE Marketplace“, Unreal Engine. Pristupljeno: 02. rujan 2024. [Na internetu]. Dostupno na: <https://www.unrealengine.com/marketplace/en-US/product/little-girl>

[28] „Mixamo“. Pristupljeno: 03. rujan 2024. [Na internetu]. Dostupno na: <https://www.mixamo.com/#/>

[29] „SuperGrid Starter pack in Environments - UE Marketplace“. Pristupljeno: 07. rujan 2024. [Na internetu]. Dostupno na: <https://www.unrealengine.com/marketplace/en-US/product/supergrid-starter-pack?sessionInvalidated=true>

6. Popis slika

Slika 1 Najigranije videoigre na platformi STEAM (Izvor: https://store.steampowered.com/charts/mostplayed)	5
Slika 2 Input Action.....	10
Slika 3 Mapiranje inputa	11
Slika 4 Locomotion igrača 1.....	11
Slika 5 Locomotion igrača 2.....	12
Slika 6 Locomotion povezivanje animacijske logike igrača 1	12
Slika 7 Locomotion povezivanje animacijske logike igrača 2	12
Slika 8 Nacrt - Ažuriranje animacija igrača 2	13
Slika 9 "Blend Space" igrača 1	13
Slika 10 Boksački stav animacije igrača 2	14
Slika 11 Nacrt pokretanja kolekcije i trgovine.....	15
Slika 12 Kolekcija igrača 1.....	15
Slika 13 Nacrt kolekcija igrača 1	15
Slika 14 Nacrt promjene oružja i prikaz na serveru i klijentu (igrač 1)	16
Slika 15 Nacrt inputa za promjenu oružja kod igrača 2	16
Slika 16 Nacrt implementacije opremanja oružjem s ograničenjem	17
Slika 17 Trgovina za igrače	17
Slika 18 Nacrt za kupovinu oružja.....	17
Slika 19 Input napada šakom	18
Slika 20 Implementacija funkcionalnosti udarca šakom	19
Slika 21 Implementacija animacija igrača 1	19
Slika 22 Animacija napada mačem igrača 1	20
Slika 23 Nacrt implementacije napada mačem	20
Slika 24 Linijsko praćenje udarca	21
Slika 25 Implementacija linijskog praćenja mača i šake.....	21
Slika 26 Inputi posebnih vještina igrača 1	22
Slika 27 Posebna vještina udarca u pod	23
Slika 28 Posebna vještina brzog trčanja	23
Slika 29 Posebna vještina besmrtnosti	23
Slika 30 Nacrt lasera 1	24
Slika 31 Nacrt lasera 2	24
Slika 32 Implementacija strelice lasera.....	25
Slika 33 Nacrt fokusa smanjenja ili uveličavanja likova za server	25
Slika 34 Nacrt smanjenja i povećanja lika za server i gosta	25
Slika 35 Nacrt inputa trčanja.....	26

Slika 36 Nacrt inputa letenja	27
Slika 37 Nacrt čučnja i proklizavanja	27
Slika 38 Animation Notify State	27
Slika 39 Događaj "AnyDamage"	28
Slika 40 Vrijednost zdravlja	28
Slika 41 Entitet srce.....	28
Slika 42 Sfera srca	29
Slika 43 Nacrt simulacije umiranja	29
Slika 44 Oživljavanje i teleportacija igrača.....	30
Slika 45 Checkpoint.....	30
Slika 46 Događaj "EventBegin" za neprijatelja	31
Slika 47 "Pawn Sensing viewport"	31
Slika 48 Nacrt implementacije "PawnSensing"-a	31
Slika 49 Nacrt brojača preživjelih neprijatelja na serveru	31
Slika 50 Nacrt za brojač živih neprijatelja.....	32
Slika 51 Dodavanje nove sekvence	32
Slika 52 Sekvence projekta	32
Slika 53 Primjer kreiranja video animacijske sekvence s parametrima	33
Slika 54 Primjer nacrt za pokretanje video sekvence	33
Slika 55 „Widgeti“ igrača 1 i 2	34
Slika 56 Događaj "Event BeginPlay"	34
Slika 57 Prilagođeni događaji za spremanje prikupljenog sadržaja videoigre.....	35
Slika 58 Putanja do datoteke "save1"	35
Slika 59 Nacrt implementacije Switch-a za događaje.....	36
Slika 60 Nacrt logike otključavanja kaveza	36
Slika 61 Nacrt ograničenja kaveza.....	36
Slika 62 Dizajn izrade sustava digitalnog lokota	37
Slika 63 Nacrt pravila lokota	37
Slika 64 Nacrt implementacije povezivanja torbice s neprijateljem	38
Slika 65 Nacrt nastajanja torbice na određenoj lokaciji i njezina sfera	38
Slika 66 Neprijatelj s ukradenom torbicom.....	38
Slika 67 Restoran za dostavu pizze.....	39
Slika 68 Nacrt iz "CityMap" svijeta za nasumično postavljanje dostave pizze	39
Slika 69 Nacrt događaja "Event Pizza BOX"	40
Slika 70 3D dizajn glavnih likova	40
Slika 71 3D dizajn sporednih likova	41
Slika 72 3D modeli oružja	41

Slika 73 Štit, mač i propeler igrača 1	42
Slika 74 Mač i laserski alat igrača 2.....	42
Slika 75 Postavljanje točke kolizije oružja neprijatelja.....	43
Slika 76 Kreiranje novčića	43
Slika 77 Materijal novčića	44
Slika 78 Kreiranje trampolina.....	44
Slika 79 Kreiranje zidnog hologramskog lasera	45
Slika 80 Kreiranje laserske hologramske ploče.....	45
Slika 81 Ostali ručno kreirani funkcionalni objekti	45
Slika 82 Primjeri "IK Rig" kostura.....	46
Slika 83 Kosturi "IK Retargetera" testnog i ciljanog modela	46
Slika 84 Animacije igrača 2.....	46
Slika 85 Klase svjetova.....	47
Slika 86 Glavni izbornik videoigre	47
Slika 87 Prozor učitavanja sesije igrača 2.....	48
Slika 88 Nacrt glavnog izbornika	48
Slika 89 Prikaz kreiranja zasluge videoigre.....	49
Slika 90 Svijet "CityMap"	49
Slika 91 Svijet "CityMap" - igralište	49
Slika 92 Svijet "CityMap" - kuća i portal starca.....	50
Slika 93 Svijet "CityMap" - garaža	50
Slika 94 Svijet "CityMap" - restoran	50
Slika 95 Svijet "Platformer" - poligoni.....	51
Slika 96 Svijet "Platformer" - laseri	51
Slika 97 Postolje.....	52
Slika 98 Nacrt teleportacije ulaska u cilj.....	52
Slika 99 Nacrt telportacije ulaska u cilj.....	52
Slika 100 Kraj sesije videoigre	53
Slika 101 Animacija trčanja i njegova implementacija „Animation Notify“	54
Slika 102 Animacija skoka lika i njegova implementacija nacrtu zvuka	54
Slika 103 Postavke podloge	54
Slika 104 Nacrt podloge	55
Slika 105 Uberduck, glasovna umjetna inteligencija	55
Slika 106 Nacrt implementacije glazbe za svijetove.....	56
Slika 107 Primjer greške prilikom stvaranja (eng. spawn) objekta na mapi	57
Slika 108 Implementacija komandi za STEAM platformu.....	65
Slika 109 Steam - prepoznavanje videoigre.....	65

7. Popis tablica

Tablica 1 Minimalni specifikacijski zahtjevi programskih alata (Izvori: [13] i [14])	7
Tablica 2 Primjer dokumentiranih funkcionalnosti i njihovih pogrešaka (Izvor:[16])	59