Izrada igre beskonačnog trčanja u programskom alatu Unreal Engine

Filip, Crnko

Undergraduate thesis / Završni rad

2018

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://urn.nsk.hr/urn:nbn:hr:211:599154

Rights / Prava: Attribution 3.0 Unported/Imenovanje 3.0

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Filip Crnko

Izrada igre beskonačnog trčanja u programskom alatu Unreal Engine

ZAVRŠNI RAD

Varaždin, 2018.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Filip Crnko

Matični broj: 43983/15–R

Studij: Informacijski sustavi

Izrada igre beskonačnog trčanja u programskom alatu Unreal Engine

ZAVRŠNI RAD

Mentor/Mentorica:

Dr. sc. Mladen Konecki

Varaždin, kolovoz 2018.

Filip Crnko

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

U ovome radu bavim se izradom igre beskonačnog trčanja, te u radu koristim alat Unreal Engine. Igra je namijenjena za korištenje na računalu, te sadržava razne elemente beskonačnog trčanja. Sama igra nema kraj, već se igrač ubrzava sa vremenom. Ima mogućnost skupljanja bodova te raznih unaprjeđenja. Igrač može sakupiti kocku koja omogućava igraču 15 sekundi zaštite protiv prepreka. Isto tako igrač ima opciju sakupiti zlatni novčić koji udvostručuje sve bodove od sakupljenih novčića unutar sljedećih 15 sekundi. Pošto se igra neprestano ubrzava, postoji lebdeći novčić koji omogućuje vračanje brzine igrača na početnu. Sama mapa na kojoj igrač igra, sastoji se od višestrukih elemenata koji se nasumično nižu, stvarajući stazu za trčanje sa višestrukim preprekama i zavojima.

Ključne riječi: Unreal Engine 4, Igra beskonačnog trčanja, UE Blueprints, generiranje mape, pojačanja (eng. *Powerups*)

Sadržaj

Sadržaj iii
1. Uvod 1
2. Metode i tehnike rada 2
3. Unreal Engine 4 3
4. Žanr igre beskonačnog trčanja 4
4.1. Temple Run 4
4.2. Subway Surfers 5
5. Mapa igre 6
5.1. Staza za trčanje 6
5.1.1. Generiranje staze 6
5.1.2. Generiranje prepreka i pojačanja 8
6. Igrač 10
6.1. Smrt igrača 10
6.2. Upravljanje igračem 13
7. Pojačanja 16
7.1. Novčići 16
7.2. Usporenje trčanja 18
7.3. Besmrtnost igrača 18
8. HUD 20
8.1. HUD unutar igre 20
8.2. Glavni izbornik i pauziranje igre 22
9. Distribucija 24
10. Zaključak 25
Popis literature 26
Prilozi 27

1. Uvod

Tema ovoga rada je igra beskonačnog trčanja. Razvoj računalnih igara je zanimljiv iz više aspekata. Kao prvo, sadrži dosta programiranja te razvija logičko razmišljanje dok sa druge strane potiče kreativnost. Isto tako sadrži razne elemente dizajna i 3D modeliranja. Sve u svemu, dok radite svoju računalnu igru imate potpunu slobodu od odabira glavne ideje do najmanjih detalja. Iako u industriji većinom radite što vam drugi kažu, uvijek postoji opcija izrade igre samostalno ili u malom timu.

Odluka za izradu računalne igre je posljedica igranja igrica od malena. Tokom pred diplomskog studija izrada računalnih igara je bila vrlo strana tema uz tek nekoliko doticaja na ponekim kolegijima. Bez ikakvog prijašnjeg iskustva, te uz potpuno ne poznavanje alata odlučio sam se za ovu temu radi osobnog izazova. Odabir ove teme natjerao me da utrošim svoje vrijeme na učenje teme koja me zanima, a za koju jednostavno nisam imao dovoljno vremena tokom studija.

Inspiracija za ovaj rad je popularna mobilna igrica „Temple Run“ [1] na kojoj sam znao provoditi slobodno vrijeme u pauzama između kolegija. Želio sam napraviti nešto slično, no isto tako nešto o čemu sam mogao naći dovoljno tutoriala pošto nisam imao nimalo iskustva.

Isprva nisam bio siguran da li je pametno da za završni rad biram temu o kojoj gotovo ne znam ništa, no radom na ovoj temi stekao sam iskustvo izrade računalnih igara, te mi je drago da sam naučio nešto novo što nije bilo dio kolegija već moj prijedlog kao tema. Isto tako smatram da današnja industrija računalnih igara je u velikom porastu, te sa sve naprednijom tehnologijom razvijaju se sve naprednije igre. To je sigurno područje u kojem bi se vrlo rado želio zaposliti nakon završetka studija iako me zanimaju i neke druge stvari.

[1] Temple Run – Popularna igra beskonačnog trčanja na Google Play platformi
https://play.google.com/store/apps/details?id=com.imangi.templerun

2. Metode i tehnike rada

Za izradu rada odlučio sam se koristiti Unreal Engine 4. Iako nisam bio upoznat sa izradom računalnih igara, nakon kratkog istraživanja Unreal Engine me je zainteresirao radi drugačijeg pristupa nego na što sam navikao. „Blueprints Visual Scripting“ je sistem korišten unutar alata koji se bazira na korištenju čvorova kao vizualnog prikaza koda. Isto kao i većina programskih jezika, služimo se objektno orijentiranim klasama prikazanim kao čvorovi. Ovakav sistem je vrlo fleksibilan jer dozvoljava dizajnerima korištenje i mijenjanje koncepata u cijelosti, što je inače dostupno samo programerima. Tako programeri stvaraju osnovu koju zatim dizajneri mogu nadograđivati (Unreal Engine Documentation, Blueprints Visual Scripting).

Unreal Engine Blueprints pruža predložak za igru u trećem licu koji sam koristio kako bi započeo sa izradom moje igre beskonačnog trčanja. Predložak sadržava igrača sa osnovnim postavkama za kretanje samoga igrača poput trčanja, skakanja te kretanja po mapi. Isto tako predložak sadržava osnovnu mapu po kojoj se može kretati te ima postavljenu kameru iza samoga igrača stvarajući pogled iz trećeg lica. Također sadrži naizgled zanemarivu, no vrlo važnu lokaciju stvaranja igrača na mapi. Sa dobivenim funkcionalnostima iz predloška lakše je početi stvarati novu igru

Kako bih krenuo sa radom bilo je potrebno naći dovoljno dokumentacije iz koje bih samostalno mogao početi učiti osnove. Unutar službene dokumentacije Unreal Engine-a pronašao sam tutorial za izradu igre beskonačnog trčanja te sam to koristio kao osnovu za rad. Tutorial sadrži sedam kratkih videa koji korak po korak opisuju izradu osnove igre. Nakon što sam završio sve tutoriale imao sam dobar uvid i osnovno znanje da nastavim dalje sam.

Za temu izgleda same igrice, od početka sam imao zamisao da bude šuma kroz koju se trči. Kako bi se ta ideja realizirala bili su potrebni 3d modeli koji nisu bili zastupljeni unutar Unreal Engine-a. Nakon što je sama funkcionalnost igre bila gotova, bilo je potrebno učitati vanjske 3d modele.

Nakon što je sama igrica vizualno i funkcionalno bila pri kraju, bilo je potrebno dodati funkcionalnost glavnog izbornika. Kao pozadinu glavnog izbornika iskoristio sam vlastito izrađenu sliku unutar Adobe Premiere Photoshop-a, isto tako izradio sam i potrebne gumbe za funkcionalnost samog izbornika te menija za pauzu.

Kao i kod svakog projekta, veliki dio vremena potrošen je na samo testiranje igrice te njezinih funkcionalnosti. Testiranje igre uzelo mi je otprilike 30% ukupnog vremena izrade, no bilo je potrebno kako bi se izbjegla većina grešaka. Za testiranje sam koristio ugrađene funkcije Unreal Engine-a za pokretanje igre u različitim modovima.

3. Unreal Engine 4

Prije nego što krenemo sa samom izradom igre, bitno je upoznati se sa korištenim alatom. Unreal Engine 4 je skup alata za izradu aplikacija u realnom vremenu. Od izrade aplikacija i vizualnih efekata do računalnih i mobilnih igara te virtualne stvarnosti. Unreal Engine 4 pruža sve što je developeru potrebno za početak, izradu i distribuciju aplikacije (Abour Unreal Engine, UE4 Features).

Unreal Engine 4 je baziran na C++ programskom jeziku. Uvijek pruža pristup cijelom kodu koji se može mijenjati i ispravljati bez ograničenja. Isto tako, pruža opciju rada sa nacrtima (eng *Blueprints*). Rad sa nacrtima ubrzava rad izrade aplikacija i igara bez potrebe rada sa samim kodom. Tako upravljamo objektima i klasama putem čvorova te stvaramo vizualnu sliku koda. Oba načina sadrže vrlo jaki debugger koji otklanja sintaktičke greške u izradi same aplikacije.

Isto tako, UE4 sadrži filmsku kvalitetu vizualnih efekata i opcija. Već samim učitavanjem početnog nivoa dobivamo visoko kvalitetne dinamične sjene, odraze i svjetlosne kanale koji zajedno stvaraju vrlo lijepu grafiku igre od samog početka. A ukoliko developer želi koristiti neke vanjske modele i efekte, vrlo lako importa vanjske objekte unutar samoga alata. Sve to moguće je testirati u bilo kojem trenutku izrade same igre pomoću build alata koji pokreću trenutnu verziju igre unutar samoga alata u nekoliko sekundi.

Jako bitno za nove developere je Unreal Engine 4 dokumentacija. Vrlo detaljni opisi i tutoriali za sve bitne teme izrade računalnih igara i ostalih aplikacija. Ukoliko developer naiđe na problem, ne treba dugo vremena prije nego što se rješenje tog problema može naći ili u službenoj dokumentaciji, ili u još opširnijoj community dokumentaciji poput tutoriala na YouTube-u. A nakon što je sama izrada igre ili aplikacije gotova, alat pruža vrlo detaljne opcije distribucije same igre. U samo nekoliko klikova developer dobiva gotov exe datoteku koju je moguće distribuirati krajnjem korisniku računalne igre.

4. Žanr igre beskonačnog trčanja

Igre beskonačnog trčanja jedne su od naj popularnijih igara na mobilnim uređajima. Ovakav žanr karakteriziran je igračem koji neprestano mora trčati kako bi ostao živ što dulje. Na putu mu se nalaze razne prepreke koje se mogu preskočiti ili zaobići. Isto tako, na samoj stazi se nalaze novčići koje igrač sakuplja, te dodatna pojačanja koja omogućuju igraču da ostane živ što dulje. Bolje verzije igara imaju staze koje se mijenjaju te sadrže razne elemente kako bi igra bila vizualno zanimljivija.

4.1. Temple Run

Jedna od popularnijih igri beskonačnog trčanja je Temple Run. U igri, ukraden je ukleti idol iz hrama te igrač sada mora bježati kako bi ostao živ. Tema staze je stari hram, koji sadrži razne prepreke i zavoje po putu. Isto tako igra sadrži pojačanja poput neuništivosti, magneta za novčiće i dvostrukih bodova.

Slika 1: Temple Run

4.2. Subway Surfers

Subway Surfers je isto vrlo popularna igra na mobilnim uređajima. Slična već spomenutom Temple Run-u, no ipak ova igra sadrži neke nove elemente. Tema igre je bježanje od inspektora i psa koji čuvaju željeznicu. Samim time, staza za trčanje sadrži puno više prepreka. Vlakovi koji se voze, prepreke po pruzi te stanice. Ali isto tako postoji puno više pojačanja koja igrač može sakupiti. Poput skateboarda ili jetpacka.

Slika 2: Subway Surfers

5. Mapa igre

U ovom poglavlju bavimo se izradom mape igre. Mapa igre beskonačnog trčanja mora se konstantno nadograđivati kako bi se igraču omogućila igra bez kraja. Tako će mo prvo govoriti o kreiranju samih dijelova mape a potom i algoritmima koji omogućuju da se ti dijelovi nasumično nižu jedan na drugoga, stvarajući stazu za trčanje. Isto tako, sama mapa mora sadržavati razne prepreke kako bi igra dobila smisao.

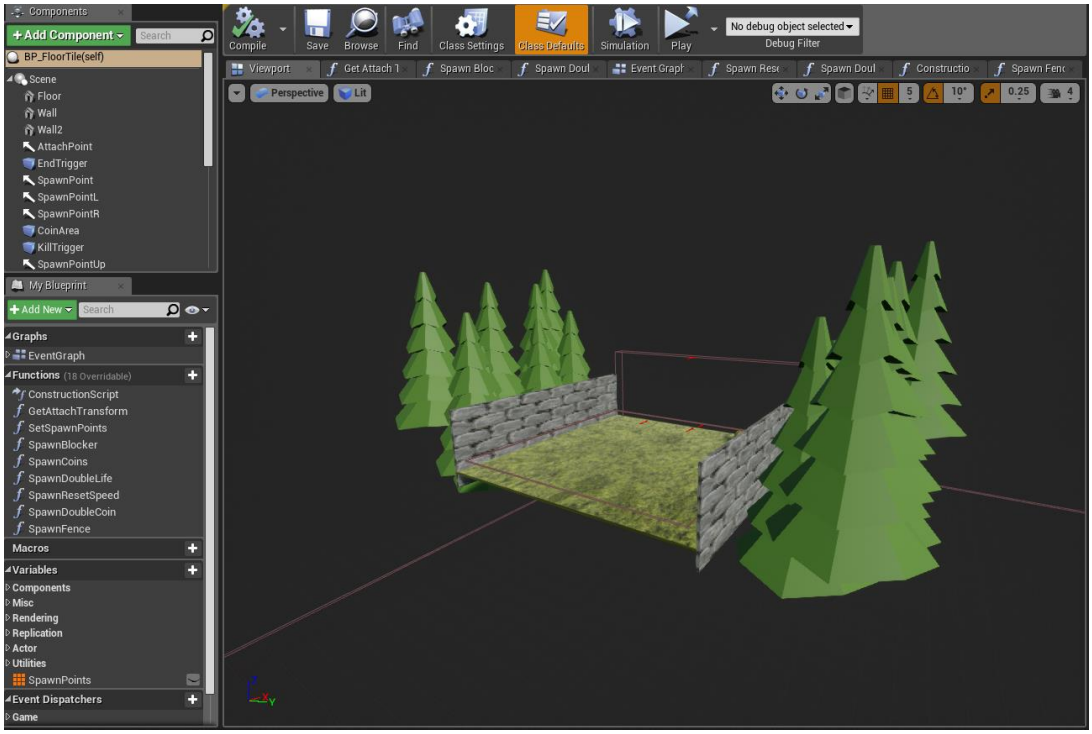
5.1. Staza za trčanje

Kako bi omogućili igraču dojam beskonačne staze za trčanje, potrebno je prvo kreirati dijelove od kojih će algoritam moći graditi stazu nizu. Unutar ovoga rada, algoritam raspolaže sa pet jedinstvenih dijelova od kojih se može nizati staza za trčanje. Prvi i naj osnovniji dio je ravna staza na kojoj se stvaraju razne prepreke, novčići i poboljšanja. Zatim slijede dva dijela koji služe za skretanje, kako bi se razbila monotonost trčanja, svako malo se ubacuje zavoj koji tjera igrača na promjenu smjera. Zadnja dva dijela su rampe prema gore i dolje, koje služe otežavanju same staze jer smanjuju vidljivost te kreiraju mogućnost ispadanja sa staze ukoliko igrač preskoči zid.

5.1.1. Generiranje staze

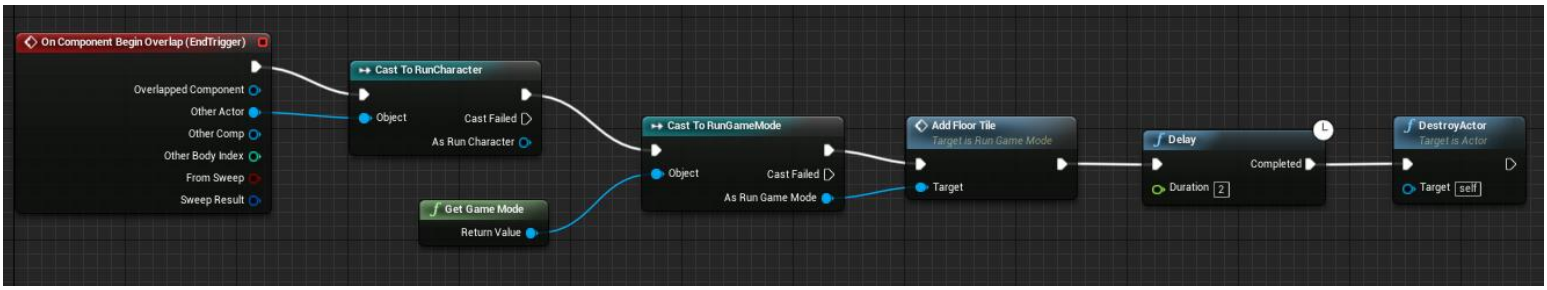
Naj bitniji dio same mape je ravna staza, te ona služi kao klasa roditelj za sve ostale elemente mape, što znači da svi elementi sadržani na ovom nacrtu biti će preslikani na sve ostale nacrte koji su dijete ove klase. Sama klasa sastoji se od vizualnih i funkcionalnih elemenata. Osnovu pruža sam pod po kojem se trči, te dva zida koji ograničavaju pad igrača van staze. Drveće iza zidova služi samo kao ukras same scene, te igrač ukoliko preskoči zid, može proći kroz drvo te pasti sa staze.

Vrlo su nam bitni funkcionalni elementi same klase. Prvo imamo četiri točke: *SpawnPoint*, *SpawnPointL*, *SpawnPointR* i *SpawnPointUp* koje služe kao lokacije za stvaranje prepreka i pojačanja do kojih će mo doći kasnije. Novčići za sakupljanje se stvaraju u grupama, pa tako se ne mogu stvarati na točci već je potrebno bilo odrediti područje u kojem se mogu stvarati, tome služi *CoinArea*. U slučaju da igrač preskoči zid staze bitan nam je *KillTrigger* koji će pozvati funkciju za ubijanje igrača te resetiranje igre.



Slika 3: Ravna staza za trčanje

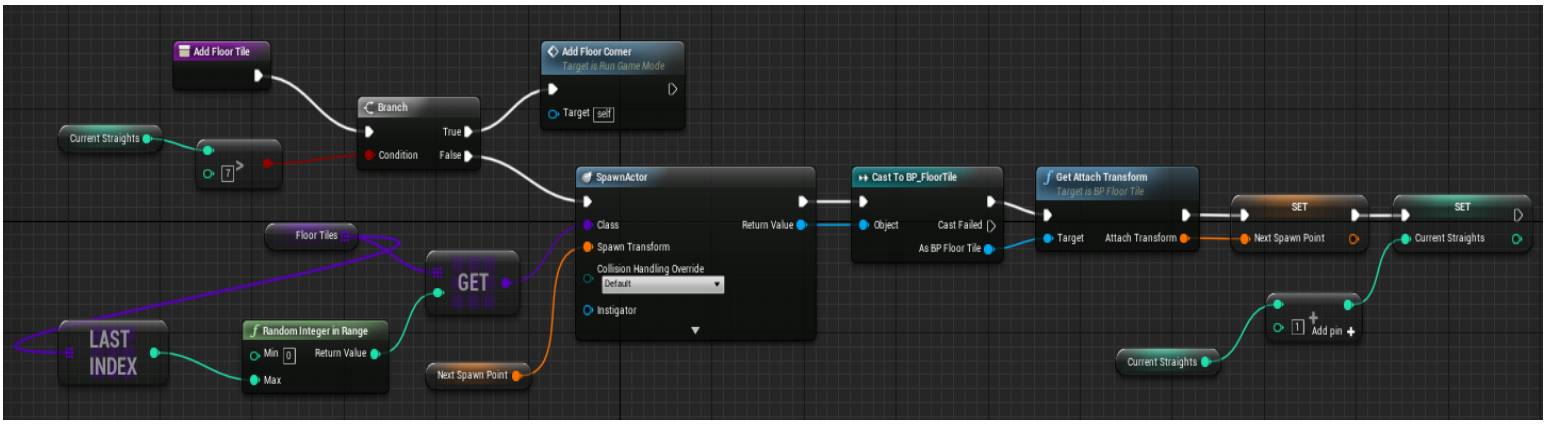
Kako bi se staza mogla neprestano generirati potreban nam je *EndTrigger* koji se nalazi na kraju svakog dijela staze te u slučaju preklapanja sa igračem poziva funkciju za stvaranje novih dijelova staze, te isto tako vrlo bitno radi optimizacije same igre, briše stare dijelove staze dvije sekunde nakon što igrač pretrči preko samog dijela.



Slika 4: EndTrigger događaj

Funkcija za kreiranje staze je vrlo kompleksna. Kao prvo, na početku igre potrebno je stvoriti određen broj dijelova, a potom je potrebno stvarati nove dijelove kako se igrač kreče. Tako svaki puta kada igrač prođe kroz *EndTrigger* funkcija prvo provjerava broj ravnih dijelova. To je važno radi variranja mape sa zavojima, ako je broj ravnih dijelova veći od sedam, funkcija će stvoriti zavoj, no ukoliko broj nije veći od sedam, funkcija će stvoriti ravni dio staze te

povećati broj ravnih dijelova za jedan. No pošto imamo više nego jedan ravni dio staze, prvo je potrebno odabrati koji dio će se stvoriti. Kako bi to omogućili, imamo polje koje sadrži sve ravne dijelove staze (rampu gore, rampu dolje te ravnicu). Funkcija nasumično odabire jedan broj te zatim stvara taj dio staze.

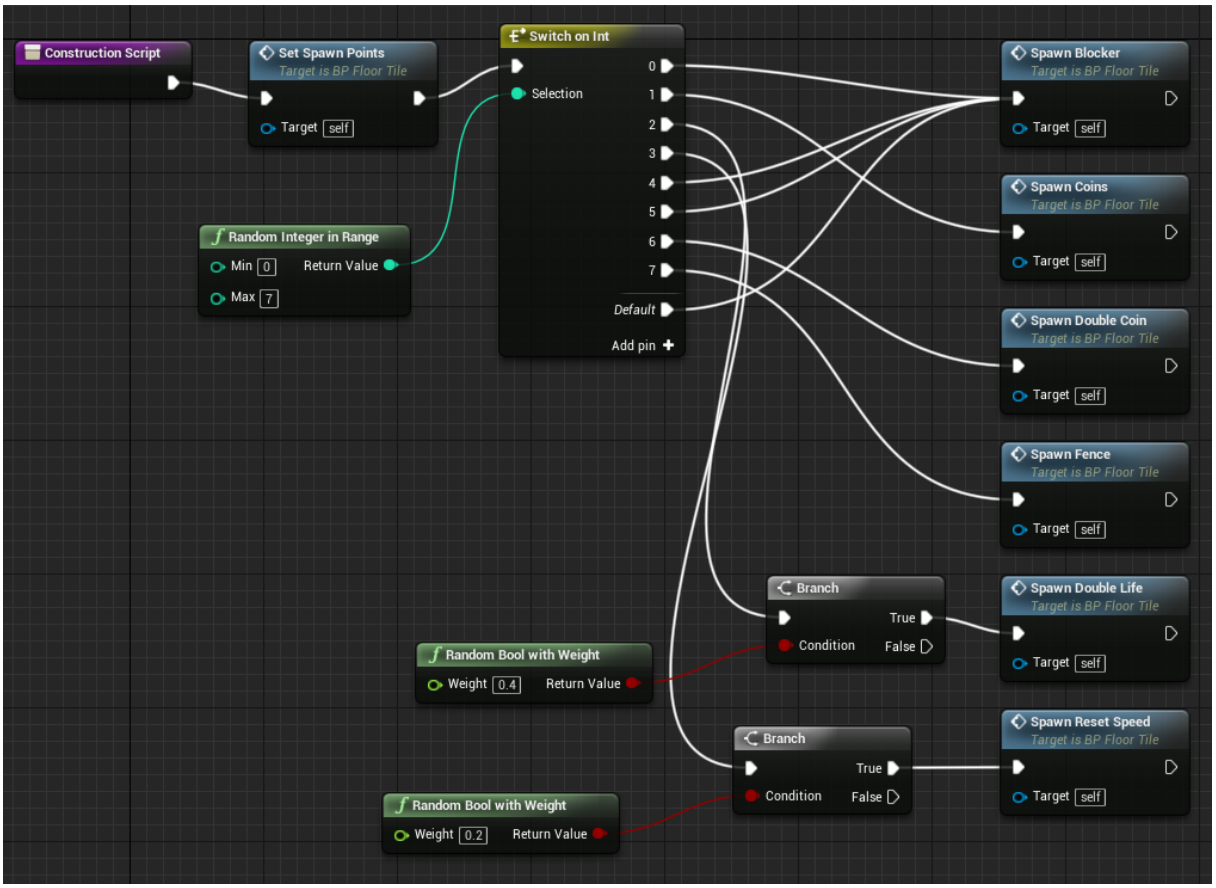


Slika 5: Kreiranje novih dijelova staze

5.1.2. Generiranje prepreka i pojačanja

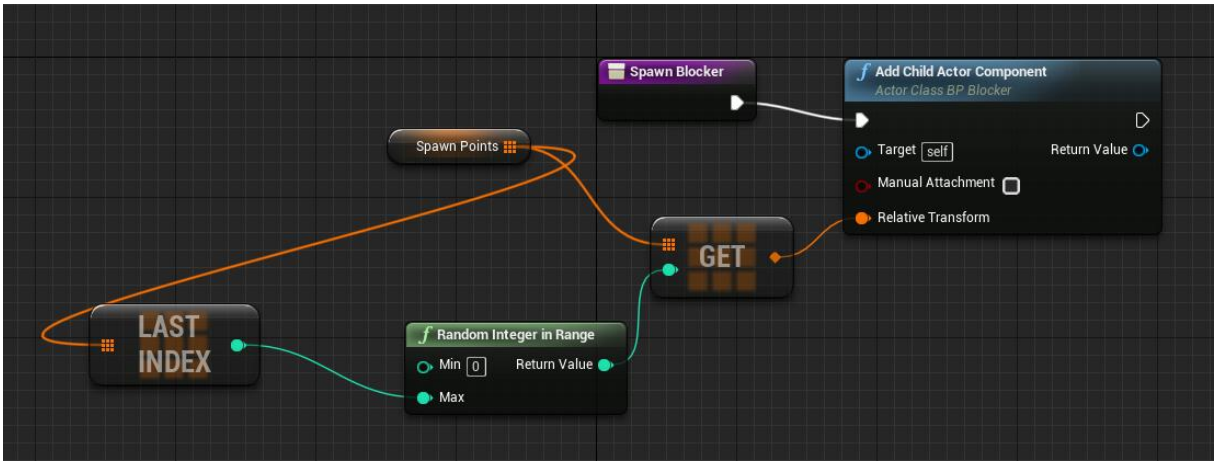
Iako je staza kao staza funkcionalna, bez prepreka i pojačanja igrica ne bi imala smisla. Tako je potrebno generirati prepreke igraču kako bi postojao način za gubljenje, te isto tako razna pojačanja koja bi motivirala i pomagala igraču da u igri napreduje dalje. Kako bi se to omogućilo potrebno napraviti skriptu koja se poziva kada se stvori ravni dio staze na kojemu želimo da se naše prepreke i pojačanja nalaze.

Kako bi to realizirao, koristio sam switch čvor koji mi omogućava različite ishode temeljem nasumičnog broja. No nije bilo dovoljno samo odabrati nasumičan broj i tako pustiti skriptu da radi, jer neke stvari u igrici trebaju biti rjeđe a neke češće. Tako sam za stvari koje želim da se češće stvaraju, dodijelio više brojeva te tako povećao šansu da onaj nasumičan broj bude jedan od dodijeljenih brojeva za željeni objekt. Slično tomu, za stvari koje želim da se rjeđe stvaraju, dodao sam petlju koja će biti istinita samo u određenom postotku pokušaja.



Slika 6: Odabir kreiranja objekta

Kada bi switch petlja odradila svoj posao, pozvala bi se funkcija koja bi kreirala željeni objekt. Funkcija bi čitala sadržaj polja sa lokacijama gdje želimo objekt da se stvori, uzimala bi nasumično lokaciju od ponuđenih kako bi se prepreke stvarale na drugim mjestima u toku same igre, te bi se potom pozivao čvor za stvaranje samoga objekta.



Slika 7: Stvaranje prepreke

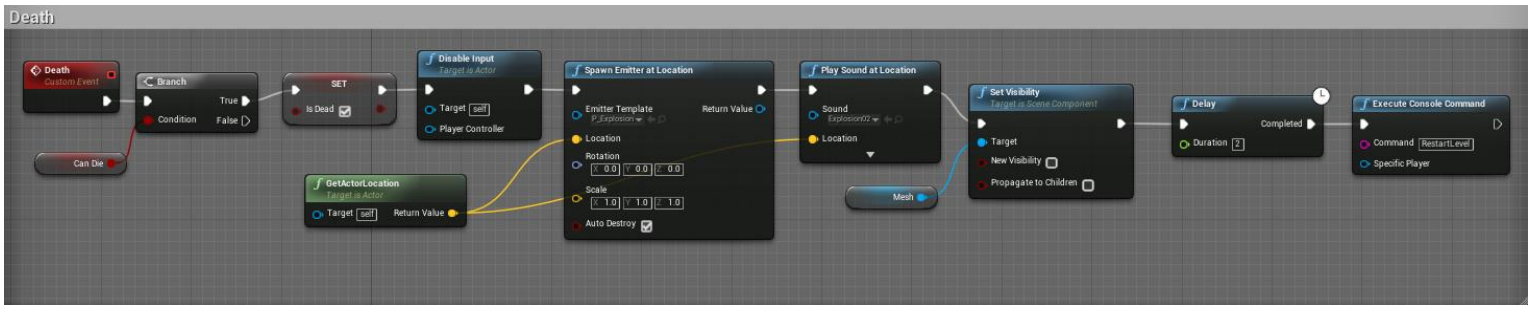
6. Igrač

U ovom poglavlju bavimo se izradom samoga igrača, lika sa kojim upravljamo. Za igru beskonačnog trčanja potrebno je nekoliko važnih stvari. Sam igrač mora neprestano trčati te ne smije nikada stati. Isto tako, kako bi igra bila zanimljivija, igrač se mora neprestano ubrzavati. Treba odrediti način na koji se igrač kreće, poput skretanja u zavojima, visina skoka i slično. I najvažniji dio, ukoliko se igrač zabije u prepreku, zid ili ukoliko iskoči sa same staze trčanja, potrebno je ubiti samoga igrača te resetirati igru.

Kada pokrenemo predložak za igru u trećem licu, automatski imamo postavljenog igrača kojeg možemo kontrolirati putem WASD kontrola, te isto tako možemo zakretati kameru putem miša. Za igru beskonačnog trčanja to ne smijemo ostaviti tako već trebamo osigurati da se kamera ne miče, te da igrač neprestano trči prema naprijed a ne da se mora držati određeni gumb za trčanje. Isto tako, bitno nam je realizirati smrt igrača u nekoliko slučajeva.

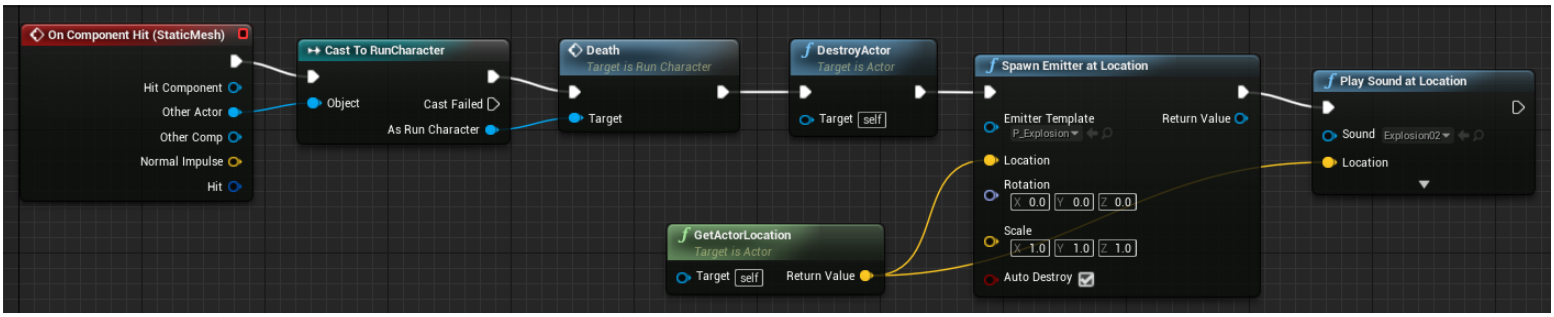
6.1. Smrt igrača

Smrt igrača realiziramo putem skripte koja prvo onesposobljava kontrole igrača kako se više ne bi moglo njime upravljati, zatim na mjestu igrača stvara vizualni efekt eksplozije, te isto tako stvara sam zvuk eksplozije. Samog igrača potrebno je sakriti, a potom sa pauzom od dvije sekunde igra se resetira na početak.



Slika 8: Smrt igrača

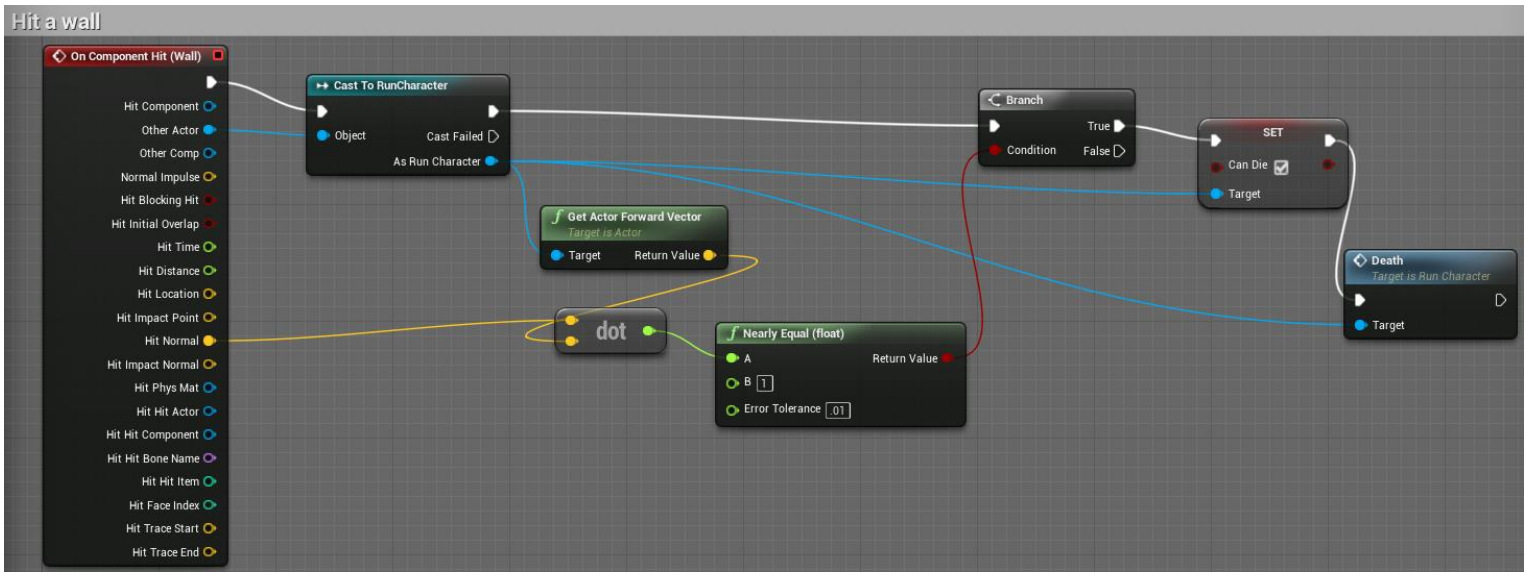
Smrt igrača potrebno je realizirati u nekoliko slučajeva. Prvi slučaj je u dodiru sa preprekom na stazi. Staza za trčanje sadrži grmlje i ograde koje se stvaraju na tri moguće lokacije na ravnoj stazi. Kako bi realizirali smrt igrača u dodiru sa preprekom, potreban nam je događaj na dodir (eng. *OnHitEvent*). Skripta zatim poziva smrt igrača objašnjenu na slici 6. Isto tako, uništava se i sama prepreka sa animacijom i zvukom. To u slučaju same smrti igrača ne igra veliku ulogu, no kasnije kada dođemo do pojačanja igrača ovo će nam biti bitno.



Slika 9: Dodir igrača sa preprekom

Drugi slučaj gdje nam je potrebno realizirati smrt igrača je sudar sa zidom u zavoju. U slučaju da igrač ne skrene, ili skrene u krivu stranu, zabiti će se u zid. Kod realiziranja ove funkcionalnosti pojavljuje se jedan problem. Svi dijelovi staze su djeca klase ravne staze, tako zidovi koji ograđuju zavoj su isti zidovi koji ograđuju ravnu stazu. Ako bi na takav zid stavili događaj koji na dodir poziva smrt, svaki puta kada bi igrač dotaknuo zid, bilo to u zavoju ili na ravnoj stazi, igrač bi umro. Očito je da to ne možemo tako realizirati. Kako bi riješili ovaj problem, kod događaja na dodir (eng. *OnHitEvent*) dodajemo još jednu provjeru. Uzimamo vektor smjera igrača, te provjeravamo da li se igrač zabio u zid direktno, ili ga je dotaknuo sa strane. Te tako realiziramo grananje, koje je istinito samo ukoliko se igrač direktno zabio u zid uz minimalnu toleranciju pogreške kako bi izbjegli računalnu pogrešku.

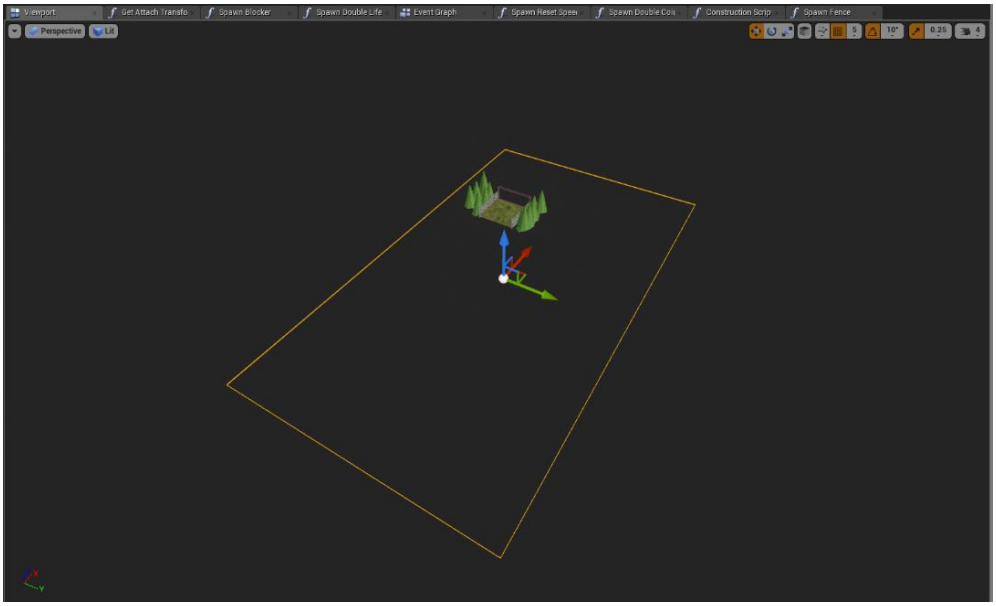
Postoji još jedan čvor prije nego što se poziva funkcija za smrt. A to je da se varijabla *CanDie* postavlja na istinu. Ta varijabla će nam kasnije služiti za realiziranje pojačanja besmrtnosti. No to pojačanje će samo djelovati na prepreke, a u slučaju da se igrač zabije u zid, on svejedno umire.



Slika 10: Sudar igrača sa zidom

Zadnji slučaj u kojem je potrebno ubiti igrača je u slučaju ispadanja sa staze trčanja. Iako staza ima zidove koji ograničavaju ispadanje sa staze, ukoliko igrač ispred rampe skoči, ima šanse ispasti sa staze trčanja. Ostajanje na stazi trčanja postaje znatno teže što se više igra ubrzava jer tako i skok igrača nosi samoga igrača dalje.

Za realizaciju ovoga problema, ispod svakoga objekta staze stavljena je zona *KillTrigger* koja je dovoljno velika da bez obzira na koji način igrač ispadne sa staze za trčanje mora dotaknuti samu zonu. Na zonu je postavljen isti događaj kao i na prepreku, te u slučaju dodira igrača sa zonom poziva se skripta za smrt igrača.

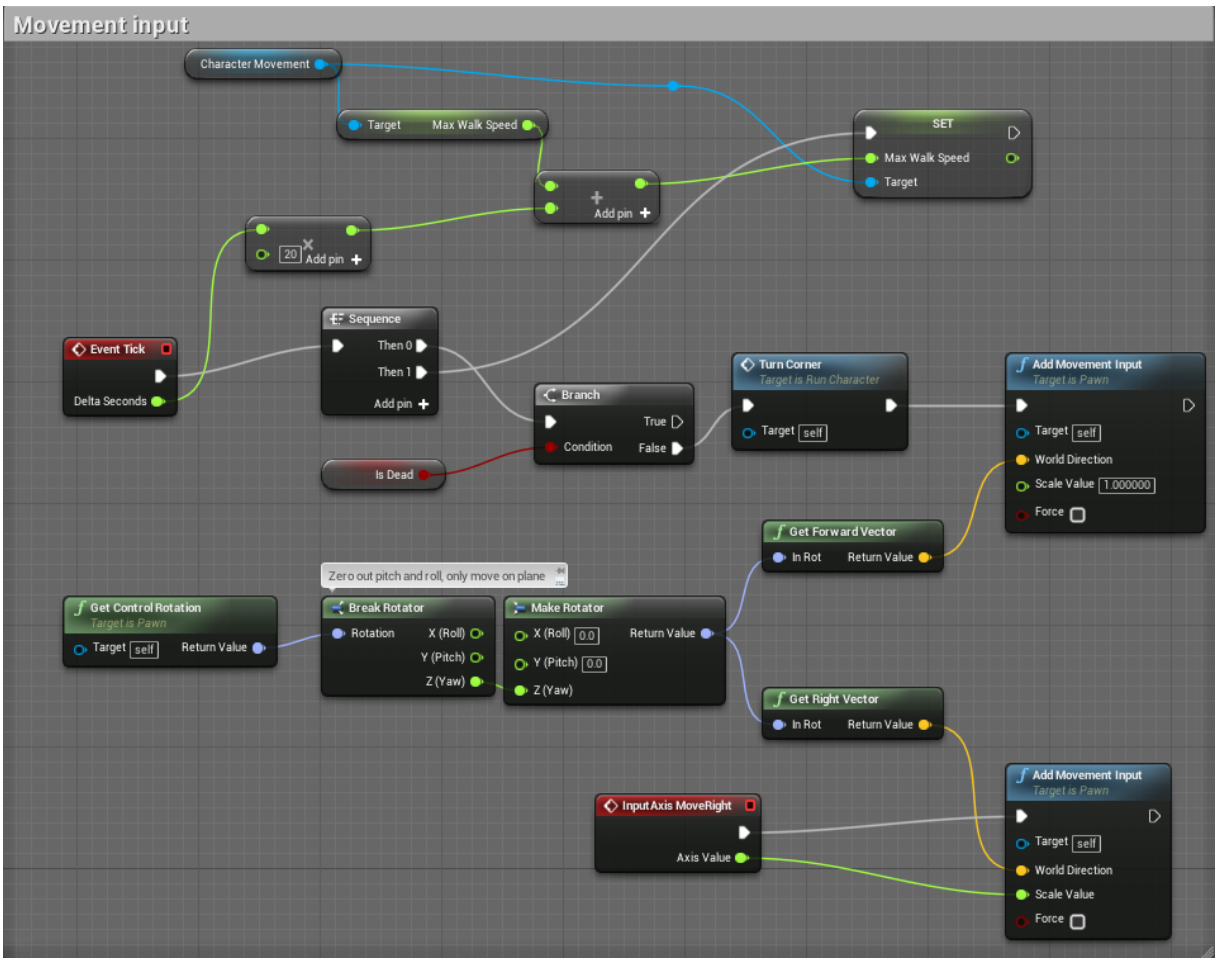


Slika 11: Zona smrti ispod staze za trčanje

6.2. Upravljanje igračem

Kako bi realizirali neprestano trčanje igrača, potrebno je zamijeniti događaj na pritisak gumba W sa događajem koji se izvršava vremenski (eng. *Event Tick*). Tako će se naš igrač neprestano kretati prema naprijed određenom brzinom, dok i dalje ostaje kontrola lijevo, desno i skok kako bi igrač mogao zaobilaziti prepreke na stazi. Vrlo važan detalj ovdje je varijabla *IsDead* iz skripte za smrt igrača, jer je potrebno staviti uvjet na neprestano trčanje ukoliko je sam igrač mrtav. Kada ne bi bilo ovoga uvjeta, u slučaju smrti igrača, zvuk i animacija bi se neprestano ponavljale jer bi se igrač neprestano zabijao u istu prepreku.

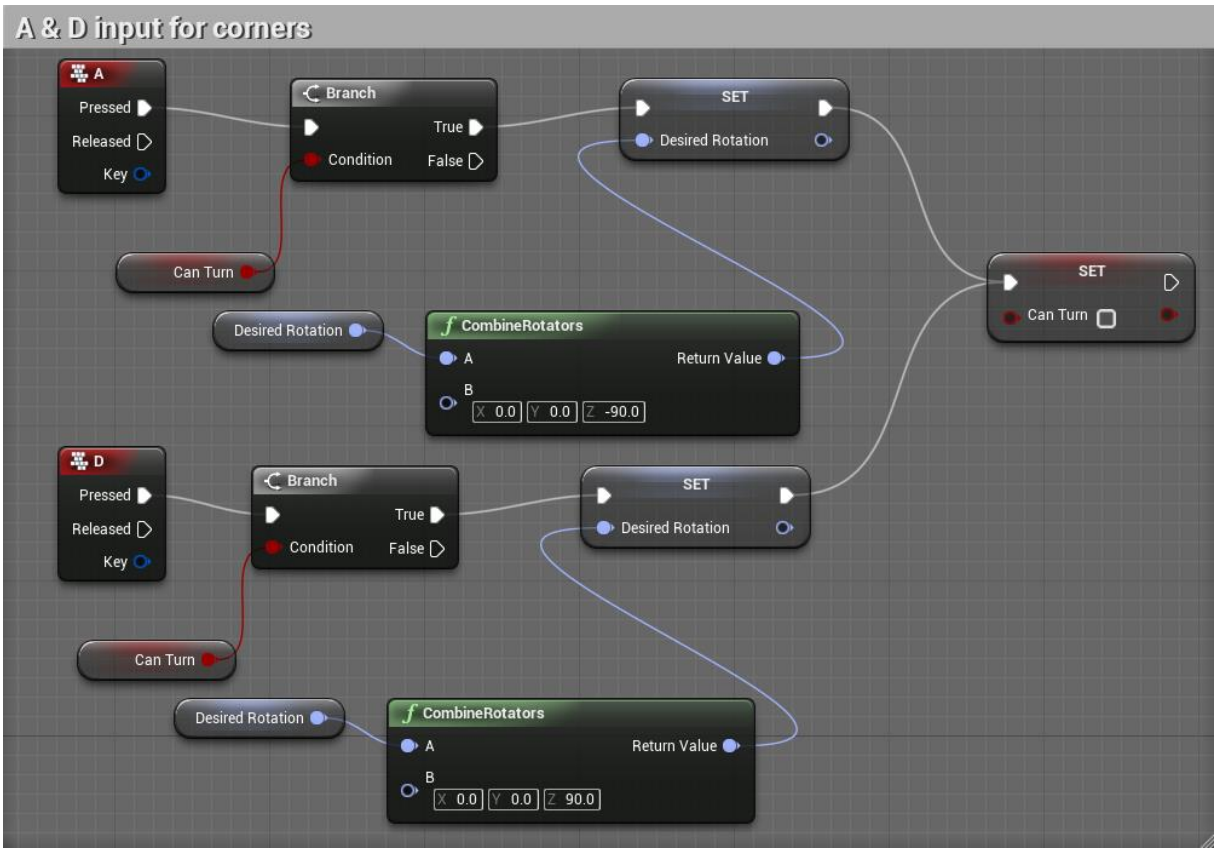
Isto tako, kao što smo već spomenuli, želimo da se igrač neprestano ubrzava. Pa tako od istog događaja koji nam pokreće igrača neprestano prema naprijed stvaramo nove čvorove koji će na svako izvođenje događaja povećati brzinu igrača za dvadeset jedinica.



Slika 12: Kretanje igrača

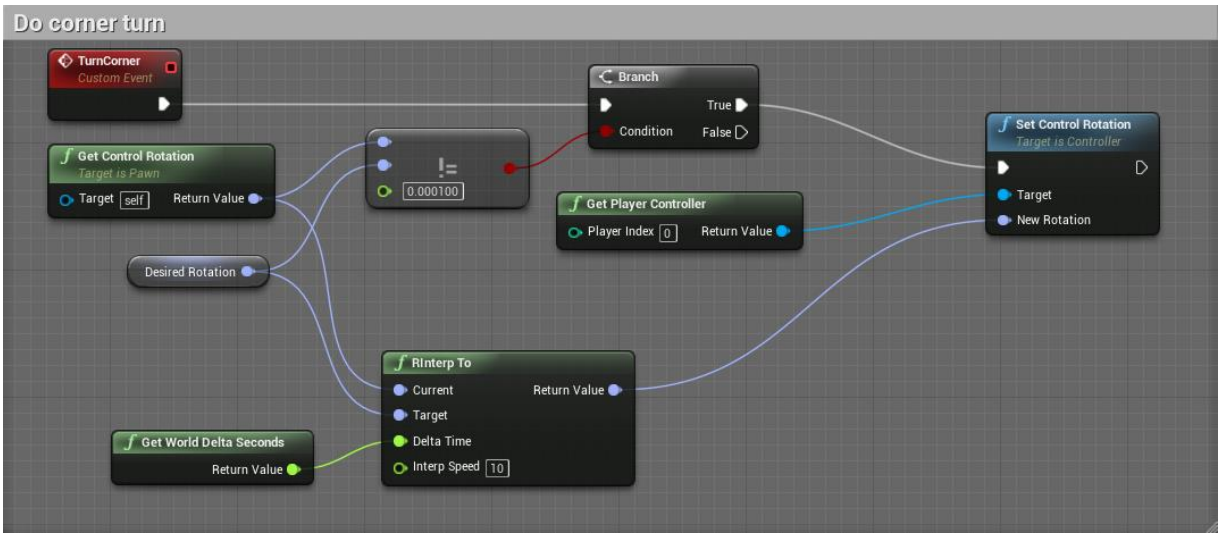
Dok tokom cijele igre, igrač ima mogućnost kretanja po stazi lijevo i desno, staza sadrži zavoje u kojima je potrebno skrenuti za 90 stupnjeva. Sa time potrebno je i rotirati kameru te promijeniti smjer kretanja. Prvo je bitno odrediti polje u kojem će takvo skretanje biti moguće. Tako za razliku od ostalog dijela staze, svaki zavoj ima *TurnZone.* Potrebno je bilo kreirati varijablu tipa boolean, koju će mo koristiti kao indikator da se igrač nalazi u zoni skretanja.

Ukoliko se igrač nalazi u zoni skretanja, naša varijabla će biti postavljena na istinu putem događaja na preklapanje igrača i zone, te će zadovoljiti uvjet petlje koja će promijeniti željenu rotaciju igrača za 90 stupnjeva u lijevo ili u desno ovisno o tipki koju smo pritisnuli.



Slika 13: Tipke A i D za skretanje u zavoju

Dok gore navedena skripta mijenja željenu rotaciju igrača, ona ne mijenja stvarnu rotaciju igrača. Za to nam je potrebna nova skripta koja će se pozivati neprestano, te provjeravati da li je trenutna rotacija jednaka željenoj rotaciji igrača. Ukoliko su trenutna i željena rotacija različite, sa minimalnim odstupanjem radi izbjegavanja računalnih grešaka, skripta će promijeniti trenutnu rotaciju igrača na željenu rotaciju postavljenu od gore navedene skripte.



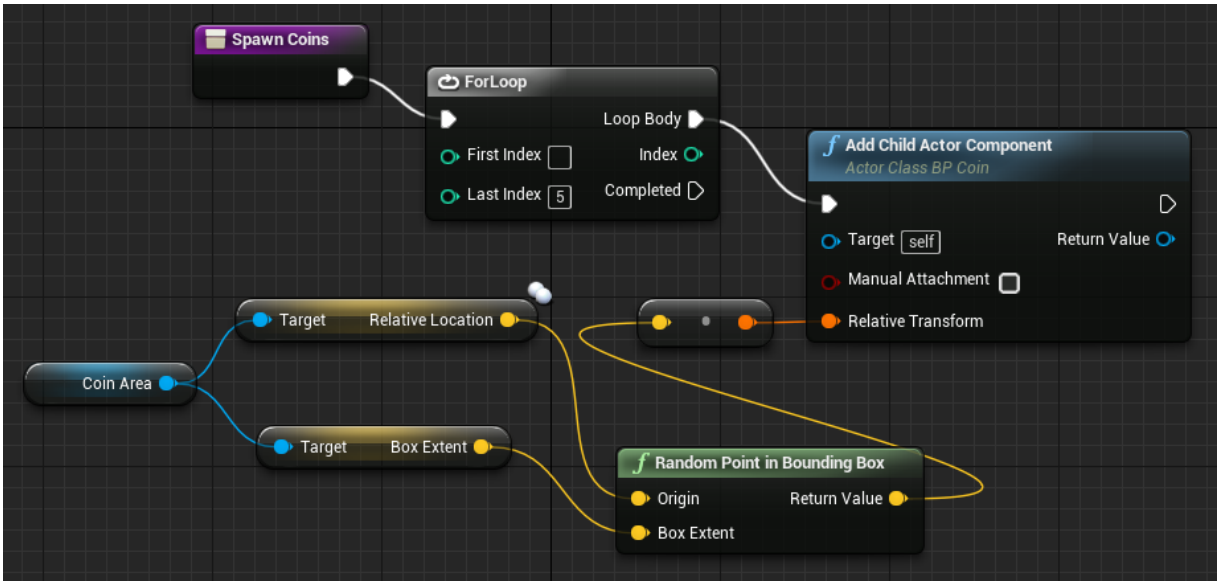
Slika 14: Skretanje u zavoju

7. Pojačanja

U ovom poglavlju baviti će mo se svime što igrač može sakupiti tokom igre, uključujući novčiće i pojačanja. Kako bi igra bila zanimljivija po stazi se stvaraju novčići koji se mogu sakupiti. Isto tako postoji zlatni novčić koji udvostručuje novčiće koje sakupljamo. Pošto staza sadrži prepreke, postoji kockica koja igrača pretvara u neuništivog, te tako igrač može prolaziti kroz prepreke. Kao što smo već objasnili prije, igra se ubrzava što dulje igrač trči. No kako bi igraču olakšali, postavljene su leteće pilule koje kada se sakupe vračaju brzinu igrača natrag na početnu.

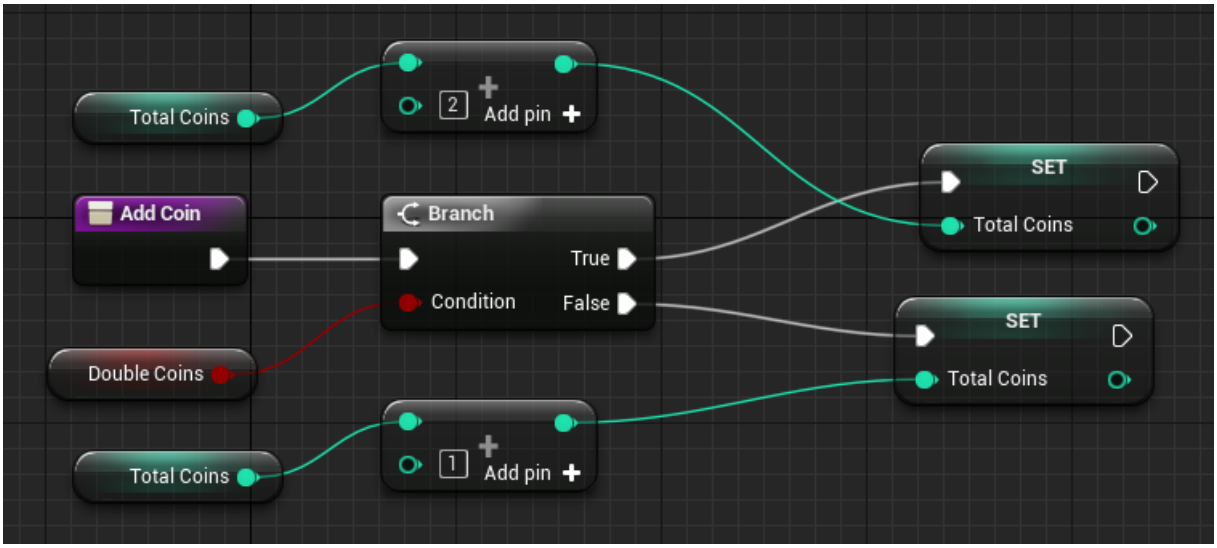
7.1. Novčići

Novčići u igri služe kao pokazatelj napretka igrača. Putem sakupljenih novčića igrač vidi koliko daleko je uspio doći. Novčići se stvaraju na ravnoj stazi u prije spomenutom polju zvanom *CoinArea.* Cilj je bio stvarati grupu novčića svaki puta na drugom mjestu, zato koristimo polje a ne točku kao što je to bio slučaj kod prepreka. Unutar polja, putem for petlje stvaramo 6 novčića.



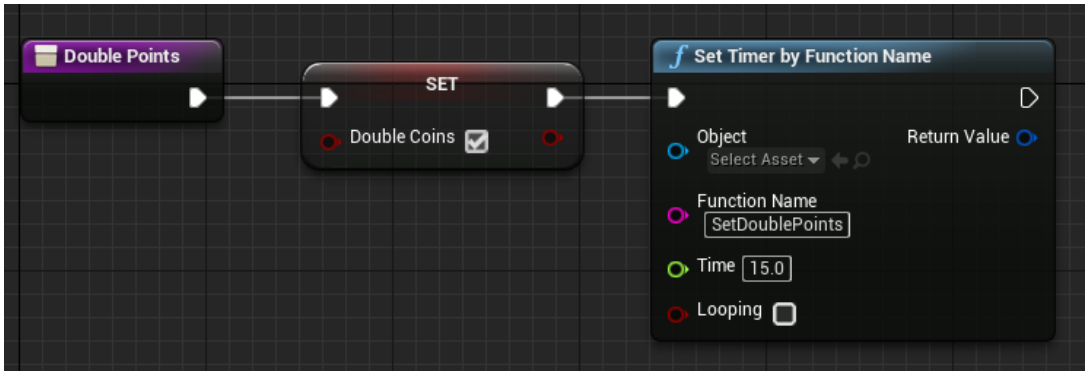
Slika 15: Stvaranje novčića u polju

Svaki novčić sadrži događaj na dodir sa igračem, koji zatim poziva funkciju za zbrajanje sakupljenih novčića. Funkcija sadrži grananje koje u jednom slučaju za svaki sakupljeni novčić ukupan broj novčića povećava za 1 ili za 2. To ovisi o vrijednosti bool varijable *DoubleCoins*.



Slika 16: Zbrajanje ukupnih novčića

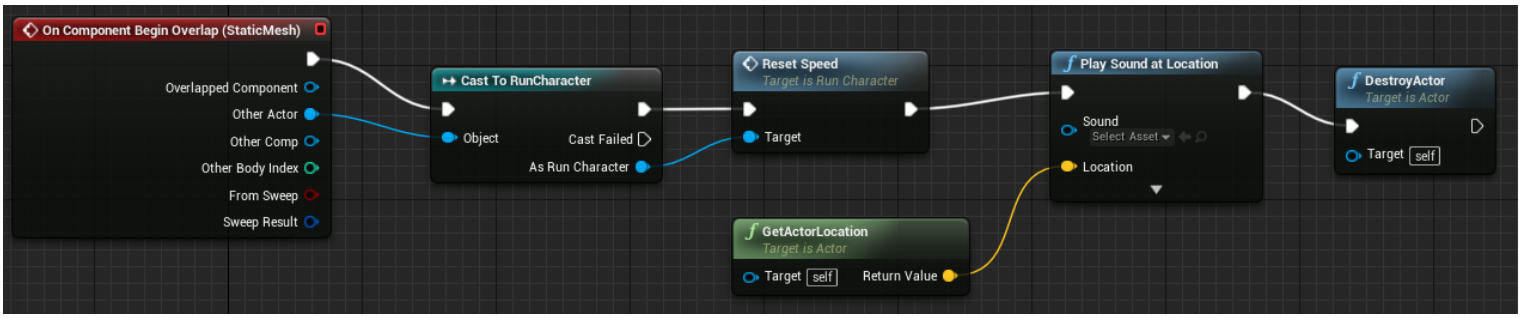
Vrijednost varijable *DoubleCoins* ovisi o zlatnom novčiću. Kada igrač sakupi zlatni novčić, vrijednost varijable se postavlja na istinu, te tako povećava zbroj svakog slijedećeg sakupljenog novčića za dva. Za petnaest sekundi, varijabla se vraća natrag u početno stanje, te u tom slučaju svi novčići vrijede kao jedan. U slučaju da igrač sakupi zlatni novčić ponovno, dok već ima jedan aktivan, vrijeme se vraća na 15 sekundi.



Slika 17: Trajanje dvostrukih bodova

7.2. Usporenje trčanja

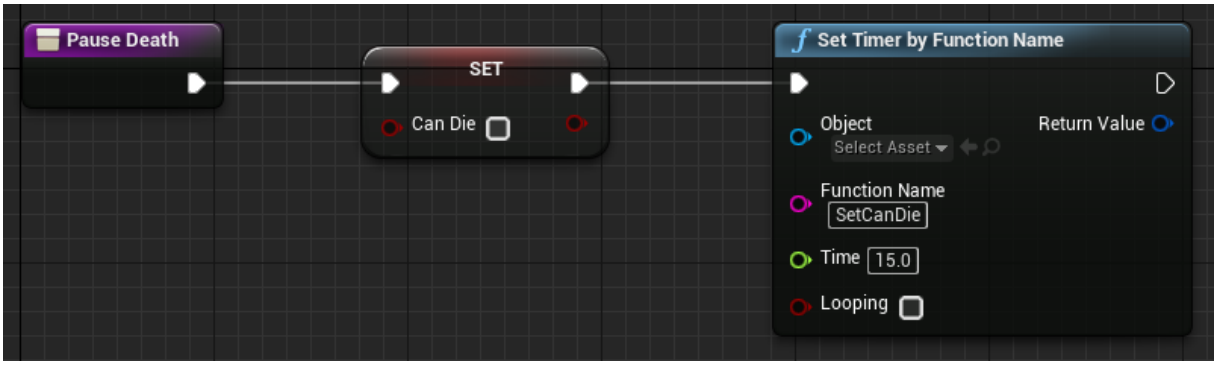
Kao što smo već više puta spominjali, igra se neprestano ubrzava. Kako bi olakšali igraču postoje lebdeće pilule koje je moguće sakupiti kako bi se brzina trčanja usporila. Ovo pojačanje je bilo lakše realizirati nego ostale. Na dodir igrača sa pilulom pozivamo funkciju koja postavlja brzinu natrag na početnu.



Slika 18: Usporenje trčanja

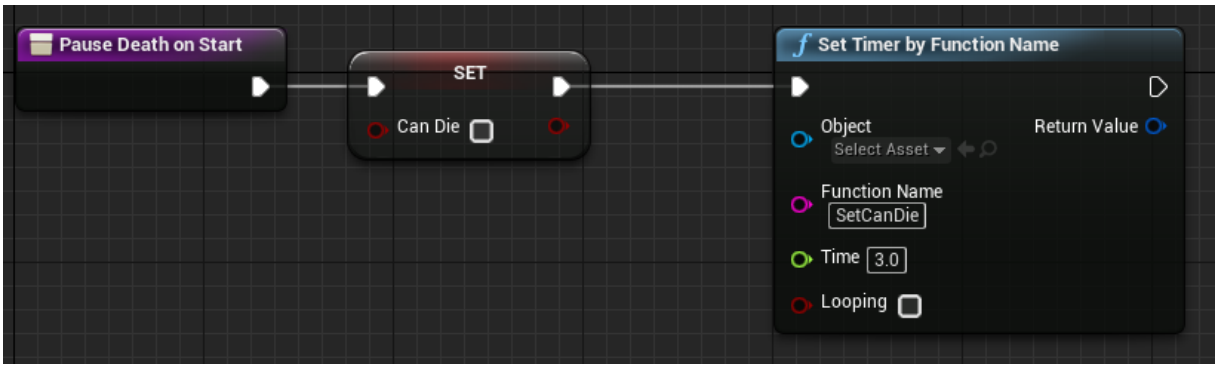
7.3. Besmrtnost igrača

Na stazi se nalazi mnogo prepreka, bilo to grmlje ili ograde. No stvaraju se nasumično i znatno otežavaju igru. Suprotno tomu, igrač ima mogućnost skupiti kockicu koja mu daje besmrtnost od prepreka na 15 sekundi. Prisjetimo se kod smrti igrača, postojalo je grananje ovisno o varijabli *CanDie*. Ukoliko je varijabla bila istinita, izvršio bi se dio koda koji bi ubio samoga igrača, no ukoliko je varijabla bila ne istinita, ne bi se desilo ništa. Tako ovo pojačanje, sve što treba raditi je promijeniti vrijednost varijable *CanDie* na ne istinu u periodu od 15 sekundi.



Slika 19: Besmrtnost igrača

Tokom testiranja same igrice otkriven je jedan problem. Često bi se znalo dešavati da igrač pokrene igricu i prije nego što ima vremena reagirati umire. Ovo bi se dešavalo zbog toga jer postoji šansa da se igrač stvori odmah ispred prepreke na stazi. Kako bi se riješio ovaj problem koristio sam već sistem koji sam imao realiziran. A to je varijabla *CanDie*. Svaki puta kada se igrica pokrene, igraču je dodijeljena besmrtnost na 3 sekunde, kako bi se izbjeglo umiranje na početku same igre.



Slika 20: Besmrtnost igrača na početku igre

8. HUD

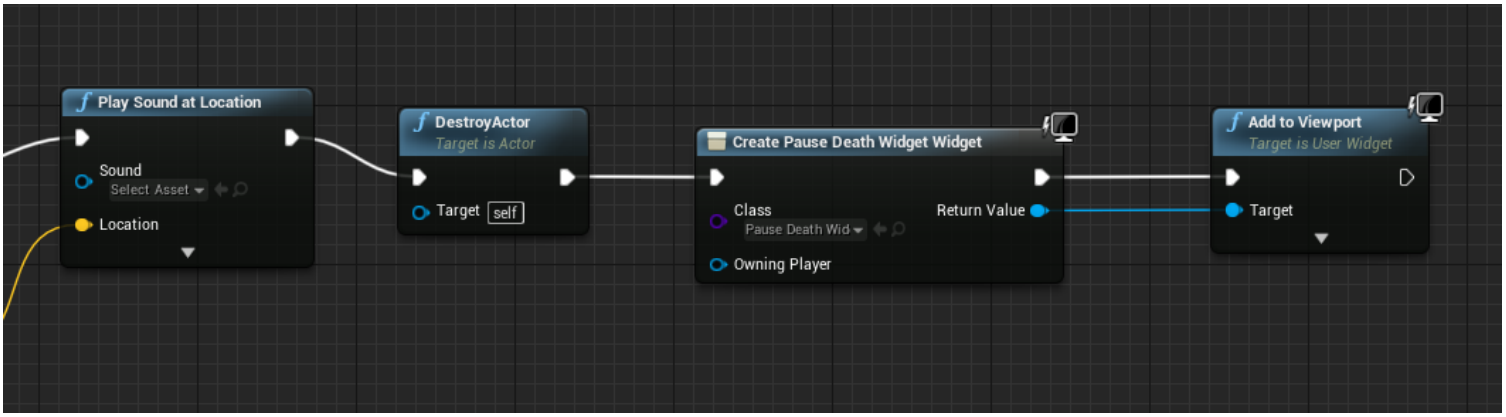
Unutar ovog poglavlja bavimo se HUD-om (eng. *Heads-Up Display*). HUD pruža informacije igraču na ekranu. Igrač ne mora gubiti fokus na stazu za trčanje, niti mora pauzirati igru kako bi vidio važne informacije o trenutnoj igri. Vrlo je važno da igrač razumije u kojem se stanju nalazi. Bez samog HUD-a, neka pojačanja bi mogla prouzročiti više problema nego pomoći. Na primjer, ako igrač sakupi besmrtnost te tako misli da se može zabijati u prepreke na stazi, kada ne bi bilo HUD-a igrač ne bi znao kada je pojačanje prestalo, te bi se tako namjerno mogao zabiti u prepreku i izgubiti igru. Isto tako, cijela igra sa sakupljanjem novčića ne bi imala smisla ukoliko igrač ne bi znao koliko je novčića sakupio. Isto tako potrebno je dodati funkcionalnost glavnog izbornika te mogućnost pauziranja igre. To će mo realizirati na sličan način kao i ostatak HUD-a, no više o tome unutar samog poglavlja.

8.1. HUD unutar igre

Prva stvar koju igrač vidi dok trči po stazi je broj sakupljenih novčića. Broj se nalazi u gornjem lijevom kutu ekrana te se prikazuje tokom cijele igre. Isto tako, u slučaju smrti igrača broj novčića se postavlja natrag na nulu. Kako bi to realizirali, potrebno je bilo napraviti HUD klasu koja će se prikazivati tokom igre. Unutar klase dodana su dva tekstualna elementa, prvi sadrži *Coins:* a drugi sadrži broj *9999* koji je bilo potrebno spojiti sa varijablom *CoinAmmount*. Tako će se taj broj dinamički mijenjati tokom igre kako se i mijenja varijabla.

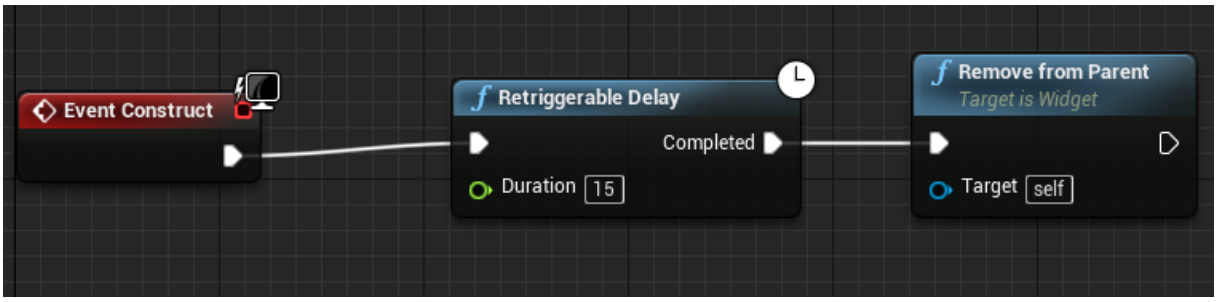
Slika 21: HUD za novčiće

Već smo govorili o pojačanjima, no važan dio pojačanja je i HUD povezan sa njima. Bitno je dati igraču povratnu informaciju o trenutnom pojačanju. Tako ukoliko igrač sakupi besmrtnost, u donjem lijevom kutu ekrana prikazati će se prikladna ikona. Isto vrijedi i za zlatni novčić koji daje dvostruke bodove. Svaka ikona ima svoju HUD klasu. Na kraju skripte za pojačanje dodan je čvor koji poziva HUD klasu te ona postaje vidljiva.



Slika 22: Pozivanje HUD klase

Nakon što je HUD klasa pozvana, prikazuje se na ekranu igrača te tako signalizira igraču da je pojačanje sakupljeno. No isto tako, potrebno je da se HUD klasa sakrije kada pojačanje istekne. To je realizirano putem samoga konstruktora HUD klase. Kada se klasa instancira, tajmer se pokreće te nakon 15 sekundi HUD klasa se sakriva sa ekrana.



Slika 23: Sakrivanje HUD klase

8.2. Glavni izbornik i pauziranje igre

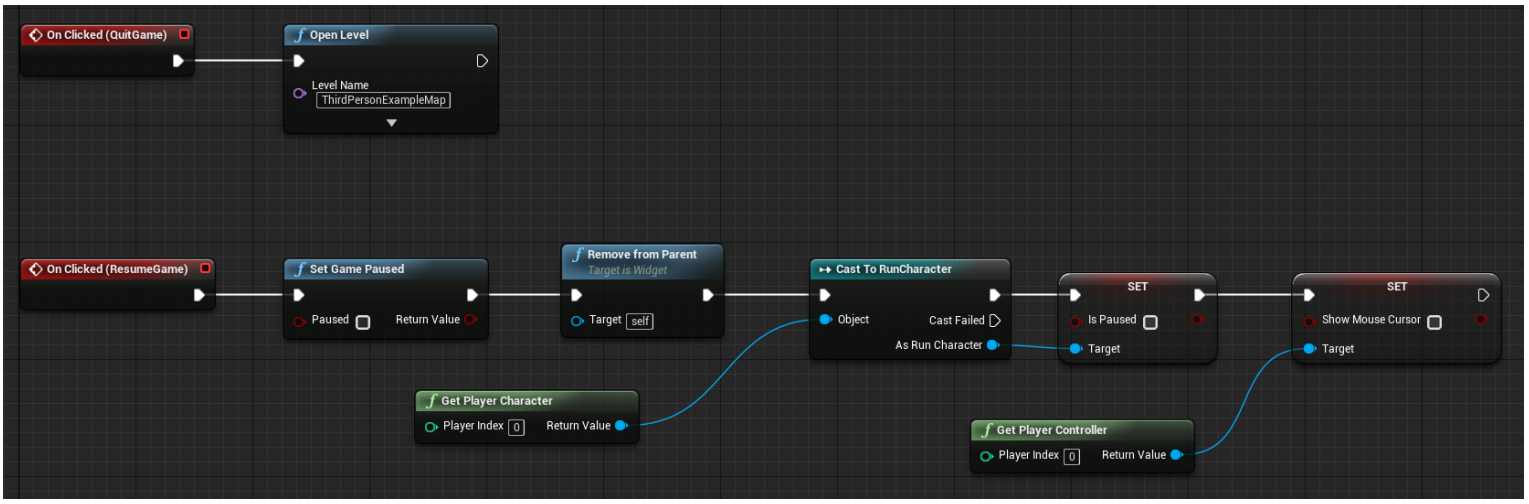
Iako je igra sama po sebi gotova, te sadrži sve funkcionalnosti potrebnu za beskonačno trčanje, fali joj interakcija sa igračem. Bez glavnog izbornika sama igra bila bi besmislena, dok opcija pauziranja je standard današnjih igara.

Kako bi realizirali glavni izbornik, potrebno je kreirati novi level. Taj level će se učitavati kada se igra pokrene dok će on pritiskom na gumb start pozivati level koji pokreće samu igru trčanja. U postavkama svijeta potrebno je postaviti novokreiranu HUD klasu kao početnu, te tako svaki puta kada se pokrene igra, prvo što igrač vidi je HUD klasa glavnog izbornika.

Slika 24: Glavni izbornik

Glavni izbornik nije samo slika, već sadrži gumbe sa svojim funkcionalnostima. Gumb *Start Game* pokreće igru na način da učitava level koji sadrži igru za beskonačno trčanje. Gumb *Options* sadrži mogućnost promjene rezolucije igre, dok gumb *Quit Game* izlazi iz same igre. Ove funkcionalnosti su vrlo jednostavne za realizirati, na pritisak gumba pozivaju se pred definirani čvorovi sa funkcijama za otvaranje novog nivoa, izlaz iz igre te promjene levela. Jedina mala komplikacija je kod samih opcija, gdje je potrebno sakriti trenutan skup gumbova, te prikazati skup gumbova sa različitim rezolucijama.

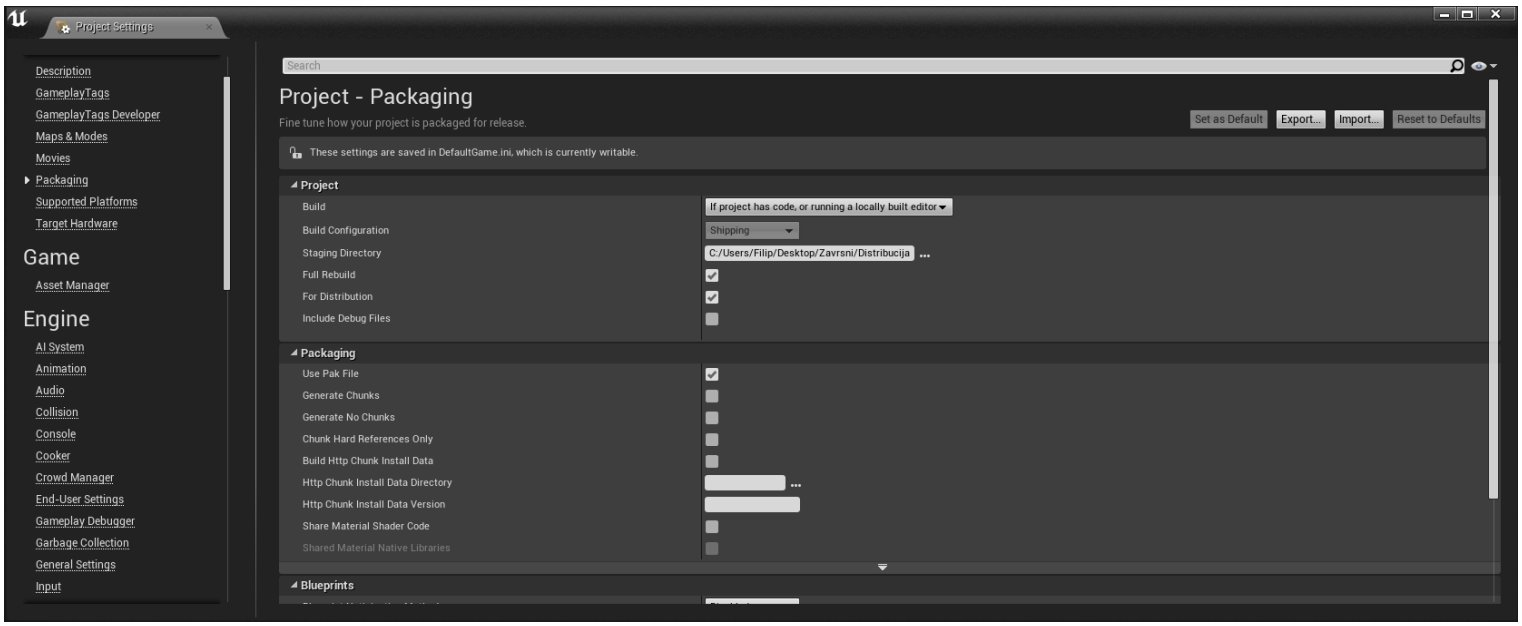
Pauziranje same igre je u postavkama levela postavljeno na gumb *Escape*. Pritiskom na gumb otvara se *PauseWidget* te se igra pauzira. Otvara se widget sa dva gumba, prvi je *Resume Game*, koji omogućuje nastavak same igre. Dok drugi gumb je *Quit Game* koji nas vrača u glavni izbornik.



Slika 25: Pauziranje igre

9. Distribucija

U posljednjem poglavlju bavimo se vrlo važnom temom. Nakon što smo zadovoljni krajnjim izgledom igre koju smo napravili, potrebno je tu igru distribuirati krajnjem korisniku. Kako bi igra bila dostupna za igranje, potrebno ju je pretvoriti u oblik koji je jednostavan za korištenje, te koji ne zahtjeva instalaciju alata za izradu same igre. Unutar Unreal Engine-a, distribucija igre dostupna je developeru u samo nekoliko klikova. Potrebno je odabrati platformu na kojoj želimo da je naša igra dostupna. Potom je potrebno ispuniti podatke o samoj igri kao što su nazivi developera, licence i slično. Isto tako potrebno je definirati koji level igra učitava na samom pokretanju te ako smo zadovoljni sa svim postavkama, Unreal Engine 4 će kreirati verziju igre dostupnu samom korisniku. Trajanje samog procesa ovisi uvelike o veličini same igre te o kvaliteti grafike na kojoj želimo distribuirati igru.



Slika 26: Distribucija igre

10. Zaključak

Izrada igre beskonačnog trčanja u Unreal Engine 4 alatu u početku se činila kao veliki izazov. No sa trudom i vremenom, uz pomoć interneta svi problemi su se brzo savladali. Prvo je bilo potrebno naći osnovu za igru. Za to sam odabrao video tutoriale iz službene UE4 dokumentacije. Nakon što sam savladao osnove same izrade igre, napravio sam kostur igre beskonačnog trčanja. Funkcije za pokretanje same igre, generiranje mape sa preprekama i novčiće za sakupljanje.

Takve igre svatko bi se brzo zasitio. Bilo je potrebno ići dalje te dodati nove detalje koji bi igru učinili zanimljivijom. Tako sam prvo krenuo sa raznim pojačanjima. Tako su nastali zlatni novčići za dvostruke bodove i kockica koja omogućuje besmrtnost. Zatim sam krenuo eksperimentirati sa samom brzinom trčanja, te sam tako otkrio kako ubrzati igrača sa vremenom, a potom i dodao pilulu koju igrač može sakupiti da se brzina trčanja vrati na početnu.

Nakon što je igra imala osnovne funkcionalnosti, vratio sam se korak unatrag te potražio 3D modele na internetu koji su bili besplatni za preuzimanje i korištenje. Te sam sa njima uredio samu stazu za trčanje i dodao novu prepreku kako bi se razbila monotonost igre. Isto tako tokom cijelog procesa potrošio sam sate igrajući samu igru te tako testirajući probleme i funkcionalnosti koje je trebalo izmijeniti ili u potpunosti maknuti.

Nakon što je igra bila u potpunosti sprema, bilo je potrebno pozabaviti se sa HUD-om, glavnim izbornikom i pauziranjem igre. Tako sam prvo utrošio nekoliko sati u photoshop-u gdje sam izradio potrebne elemente glavnog izbornika i samoga HUD-a. Kada je to bilo gotovo, premjestio sam slike unutar samoga UE4 programa, te sam izradio potrebne klase. Trebalo je nekoliko dana da savladam kako upravljati izbornicima i HUD-om unutar samoga UE4 programa.

Sve u svemu, cijeli projekt mi je otvorio nove prilike da savladam izradu računalnih igara. Iako je projekt zahtijevao jako puno utrošenog rada i truda bilo je vrijedno vremena i naučenih vještina.

Popis literature

[1] Unreal Engine Documentation, „Blueprints Visual Scripting“, [Na internetu]. Dostupno na: https://docs.unrealengine.com/en-us/Engine/Blueprints. [Pristupljeno: 02-Kol-2018].

[2] Unreal Engine Documentation – Video Tutorials, „Blueprint Endless Runner“, [Na internetu]. Dostupno na: https://docs.unrealengine.com/en-us/Videos/PLZlv_N0_O1gbY4FN8pZuEPVC9PzQThNn1. [Pristupljeno: 10-Velj-2018]

[3] About Unreal Engine „Unreal Engine Features“, [Na internetu]. Dostupno na: https://www.unrealengine.com/en-US/features. [Pristupljeno: 31-Kol-2018].

[4] TurboSquid, „PigArt Tree“, [Na internetu]. Dostupno na: https://www.turbosquid.com/FullPreview/Index.cfm/ID/1038262. [Pristupljeno: 20-Srp-2018].

[5] TurboSquid, „Fence wooden“, [Na internetu]. Dostupno na: https://www.turbosquid.com/3d-models/free-fence-wooden-wood-3d-model/876967. [Pristupljeno: 22-Srp-2018].

[6] Imangi Studios, „Temple Run“, [Na internetu]. Dostupno na: https://play.google.com/store/apps/details?id=com.imangi.templerun. [Pristupljeno: 31-Kol-2018].

[7] Kiloo, „Subway Surfers“, [Na internetu]. Dostupno na: https://play.google.com/store/apps/details?id=com.kiloo.subwaysurf. [Pristupljeno: 31-Kol-2018].

Prilozi

[1] Igra beskonačnog trčanja, „MyProject“ [Unreal Engine Project File] . Dostupno na: https://drive.google.com/drive/folders/1tESTAPWwC88wkrdZc-nwNSbT_QMsYkOw?usp=sharing

[2] Distribucija igre, „Igra beskonačnog trčanja“. Dostupno na: https://drive.google.com/file/d/1bz4w4kBZu_XMSXv-E1vSale06VIWSTaQ/view?usp=sharing