



Kauno technologijos universitetas

Informatikos fakultetas

Pirmojo asmens šaudyklė virtualiojoje realybėje

Virtualiosios realybės technologijų semestro projektas

Martynas Kniukšta

Paulius Šinkūnas

Projekto autorius (-iai)

lekt. Andrius Paulauskas

Priėmė

Kaunas, 2022

Turinys

Lentelių sąrašas.....	3
Paveikslų sąrašas.....	4
Įvadas.....	5
1. Žaidimo idėja.....	6
2. Technologijos.....	9
3. Realizacija.....	13
Rezultatai ir išvados.....	17

Lentelių sąrašas

No table of figures entries found.

Paveikslų sąrašas

Pav. 1.1 "Killing floor 2" žaidimas.....	6
Pav. 1.2 Mūsų realizuotas žaidimo pavyzdys.....	7
Pav. 1.3 Žaidžiamas žaidimas. Matomas bangų ir priešų skaičius.....	8
Pav. 1.4 Veikėjo modelis su sekimo sensoriais.....	8
Pav. 2.5 Ginklo blykstės medžiagos grafas.....	9
Pav. 2.6 Pagrindinė žaidimo progreso logikos.....	10
Pav. 2.7 Ginklo šaudymo logikos grafikas.....	10
Pav. 2.8 Mūsų pačių kurta C++ klasė.....	11
Pav. 2.9 Garso variacijos grafas.....	11
Pav. 3.10 "Gameplay Framework" klasės (mėlynai pažymėtos „Unreal Engine“ klasės).....	13
Pav. 3.11 Ginklo kulkos medžiagos grafas.....	14
Pav. 3.12 Žolės medžiagos grafas.....	15
Pav. 3.13 Perlino garso tekstūra.....	16

Įvadas

Virtualiosios realybės technologija nėra itin nauja. Vystytis pradėjo jau XX a. viduryje, tačiau labai stiprus technologijos progresas yra pastebimas nuo 2010 m. Būtent nuo šių metų kelios technologijų įmonės susidomėjo šios technologijos potencialu ir pasiūlė ne vieną prototipą, o galiausiai ir produkcinis įrenginių variantus. Tolimesnėje perspektyvoje galima numatyti, kad virtualiosios realybės įrenginiai taps vis labiau prieinami eiliniams naudotojams, o ne tik entuziastams.

Sričių, kuriose būtų taikoma virtualioji realybė, netrūksta. Dažniausiai ji taikoma medicinos, apmokymų, architektūros, industrinio projektavimo, pramogų ir kitose srityse. Pramogų sritis turi daug potencialo ir nors pastaraisiais metais buvo išleisti keli didesni virtualiosios realybės žaidimai, tokie kaip „Half-Life: Alyx“, „Boneworks“, „The Walking Dead: Saints & Sinners“, tačiau kol kas kokybiškų virtualios realybės žaidimų pasiūla nėra labai didelė. Pasiūlą būtų galima praplėsti.

Kuriant kompiuterinius žaidimus svarbu, kad jie būtų intuityvūs, įdomūs, estetiškai pateikti ir turėtų kokių nors naujovių. Naujovės gali būti perteikiamos pasitelkiant naujas žaidimų mechanikas, pasakojamas istorija, panaudojant naujas technologijas. Naujovės gali lengviau pritraukti naudotojų dėmesį ir padidinti žaidimo pardavimų kiekį.

Darbo tikslas – papildyti virtualiosios realybės žaidimų pasiūlą sukuriant naują žaidimą, kuriame būtų naudojama inovatyvi valdymo schema. Tikslui pasiekti iškelti šie uždaviniai:

1. Sukurti žaidimo koncepciją
2. Pasirinkti tinkamiausias technologijas virtualiosios realybės žaidimo kūrimui
3. Suprojektuoti žaidimą ir jį realizuoti

1. Žaidimo idėja

Šiame skyriuje pateikiama aprašyta žaidimo idėja.

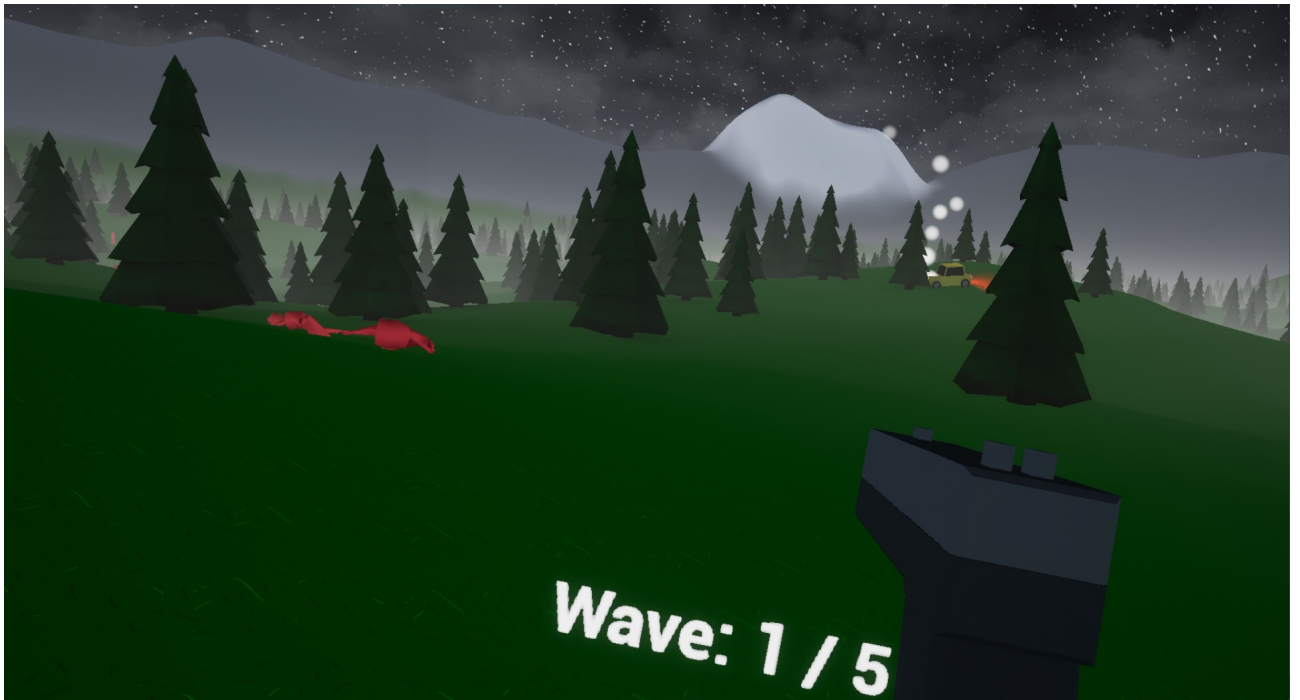
1.1. Žaidimo mechanikos

Kuriamas žaidimas yra pirmojo asmens šaudyklė. Žaidime reikia nušauti tave bandančius nužudyti priešus norint toliau progresuoti, žaidžiama vieno asmens. Priešai puola bangomis su mažais laiko tarpais pasiruošti ar susirasti naują vietą žemėlapyje. Mūsų žaidimas stipriai įkvėptas žaidimo „Killing Floor 2“ pagrindinių žaidimo mechanikų.

Žaidėjas valdo veikėja su kuriuo turi bandyti išgyventi per visas priešų bangas. Žaidėjui neišgyvenus žaidimas laikomas kaip pralaimėtas. Kitu atveju žaidėjas laimi.



Pav. 1.1 "Killing floor 2" žaidimas.



Pav. 1.2 Mūsų realizuotas žaidimo pavyzdys

Žaidime yra pagrindinio veikėjo givybes, tačiau jos žaidėjui nerodomos norint padidinti įtampą žaidime. Po kiekvienos bangos yra 30 sekundžių tarpas pasiruosti naujai bangai. O kiekvienoje naujoje bangoje atsiranda vis daugiau priešų ir jie atsiranda greičiau.

1.2. Naudotojo sąsaja

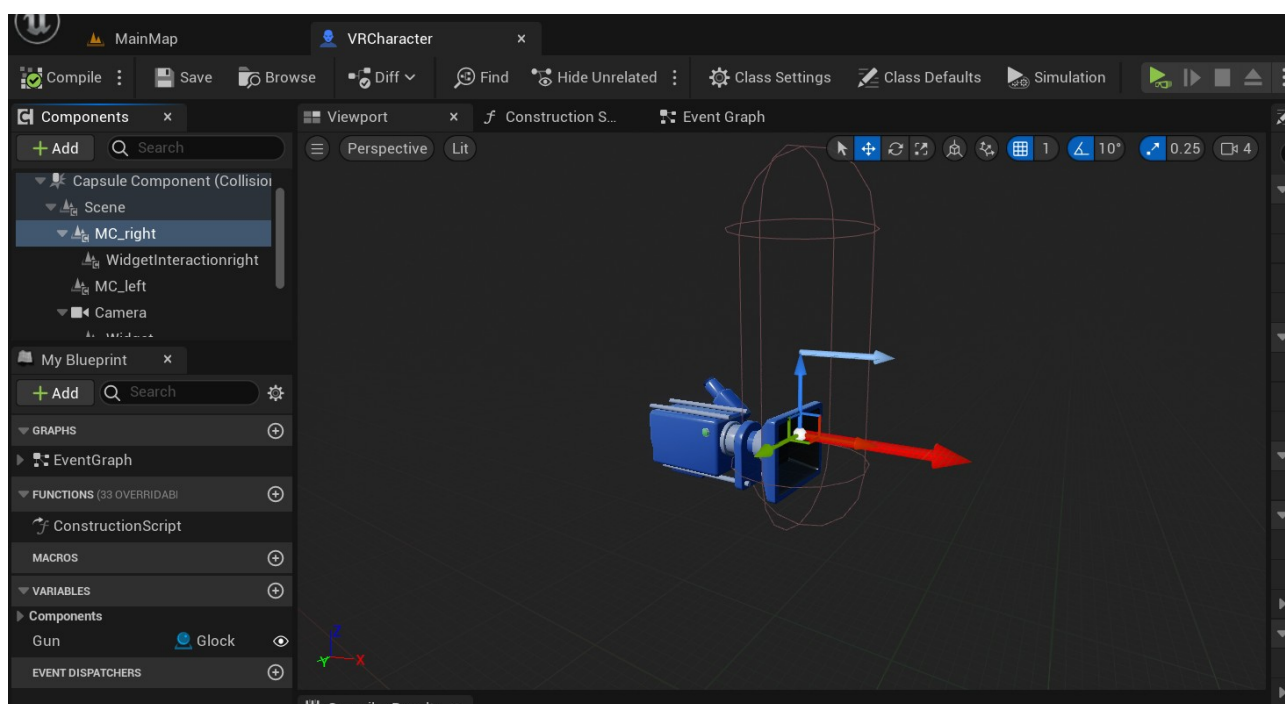
Žaidėjas valdo veikėją su Oculus quest 2 valdikliais, pagrindinis valdymas vyksta su dešne ranka, su kuria yra laikomas ginklas ir iš jos šaudoma, tai pat meniu pasirinkimai valdimi ta pačia ranka. Prie žaidėjo ginklo yra parodomo aktuali informacija, tai dabartinė banga, visu bangu skaičius ir likęs priešų skaičius bangoje.



Pav. 1.3 Žaidžiamas žaidimas. Matomas bangų ir priešų skaičius

1.3. Valdymas

Žaidėjas valdo virtualų veikėją, kuris naudojamas priešams šaudyti. Virtualus veikėjas seka judesio sekiklio judesius. Su virtualios realybės valdikliais valdomos kaire ir dešinė rankos. Su kuriomis ir vyksta pagrindinis žaidimo veiksmas.



Pav. 1.4 Veikėjo modelis su sekimo sensoriais

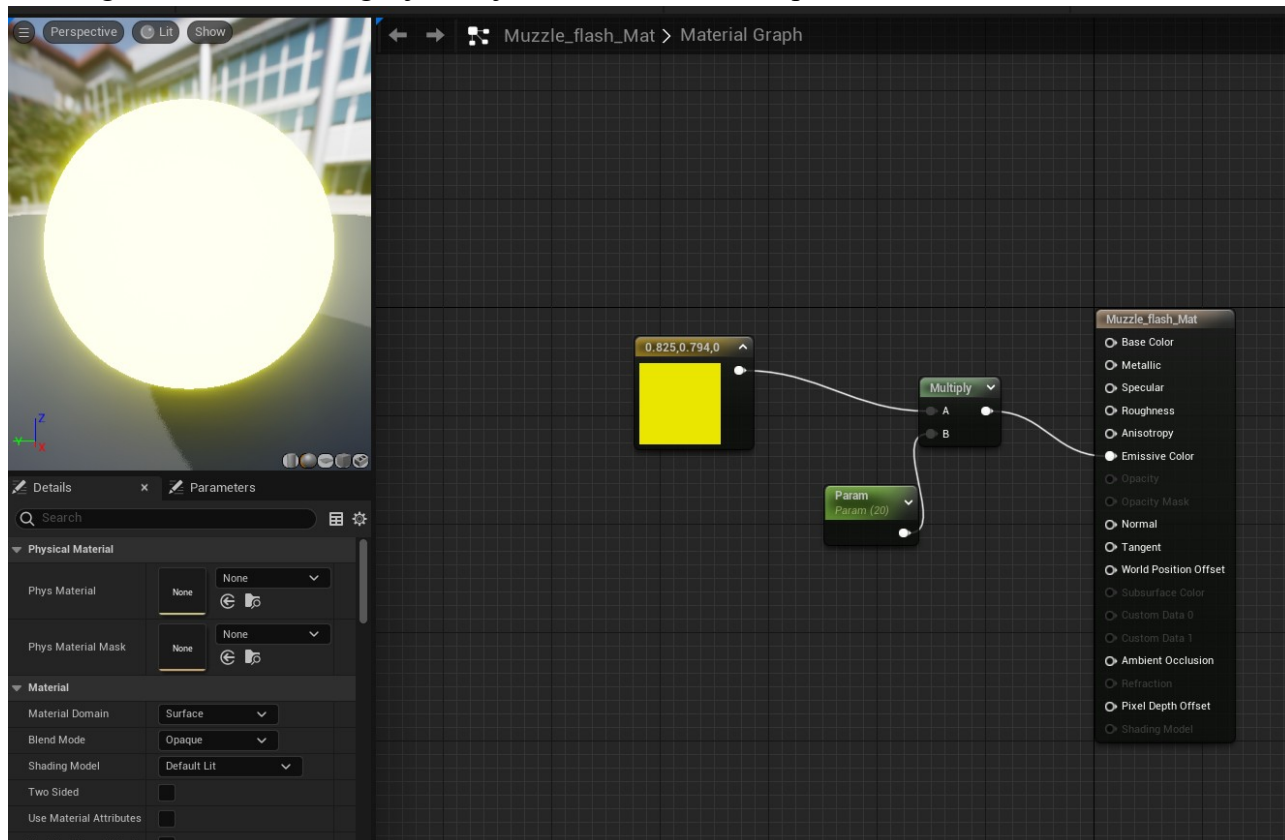
2. Technologijos

Šitame skyriuje pateikiamos technologijos, kurios buvo naudojamos projektui realizuoti.

2.1. Žaidimų kūrimo variklis

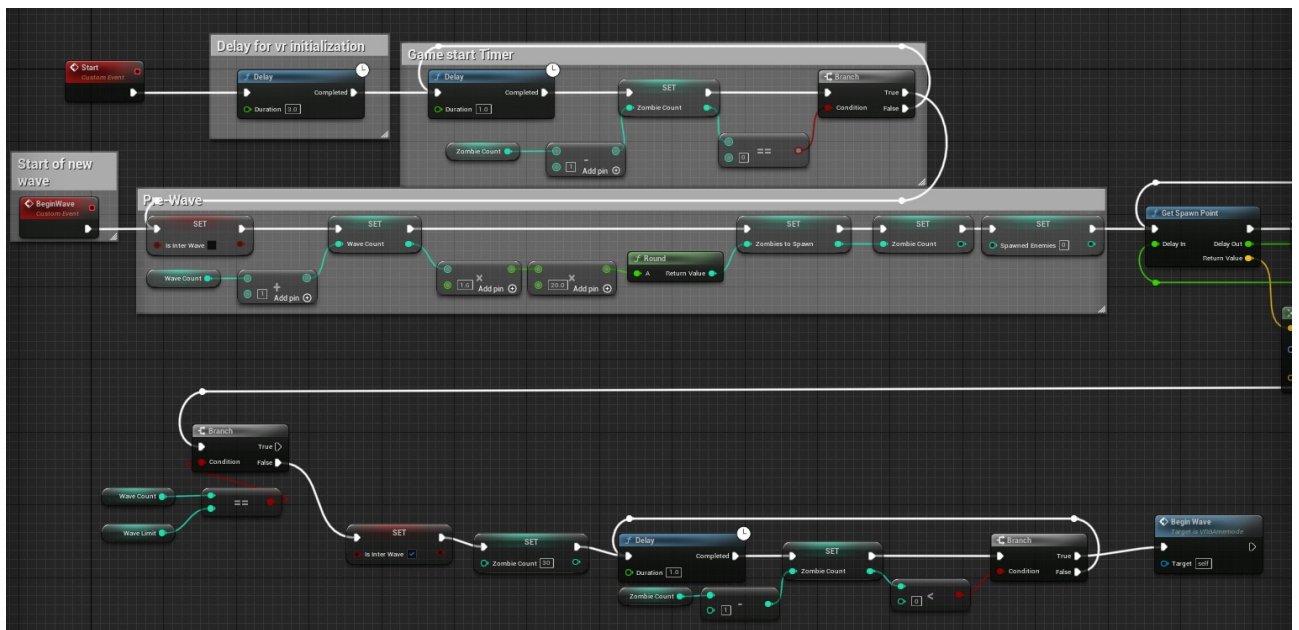
Žaidimas kurtas naudojant „Unreal Engine“ variklio 5.0.3 versiją. Žaidimų variklis turi didelį integraciją su didžiąja dalimi VR platformomis. Tai leidžia sutaupyti laiko, jeigu reikėtų kompiliuoti žaidimą kelioms sistemoms („Oculus“, „Steam VR“ ar „Windows Mixed Reality“).

Taip pat pravertė žaidimų variklio medžiagų redagavimo įrankis, kuriuo buvo kuriamos įvairios medžiagos efektams ir redaguojant objektų teksturas, taikant prie žaidimo.

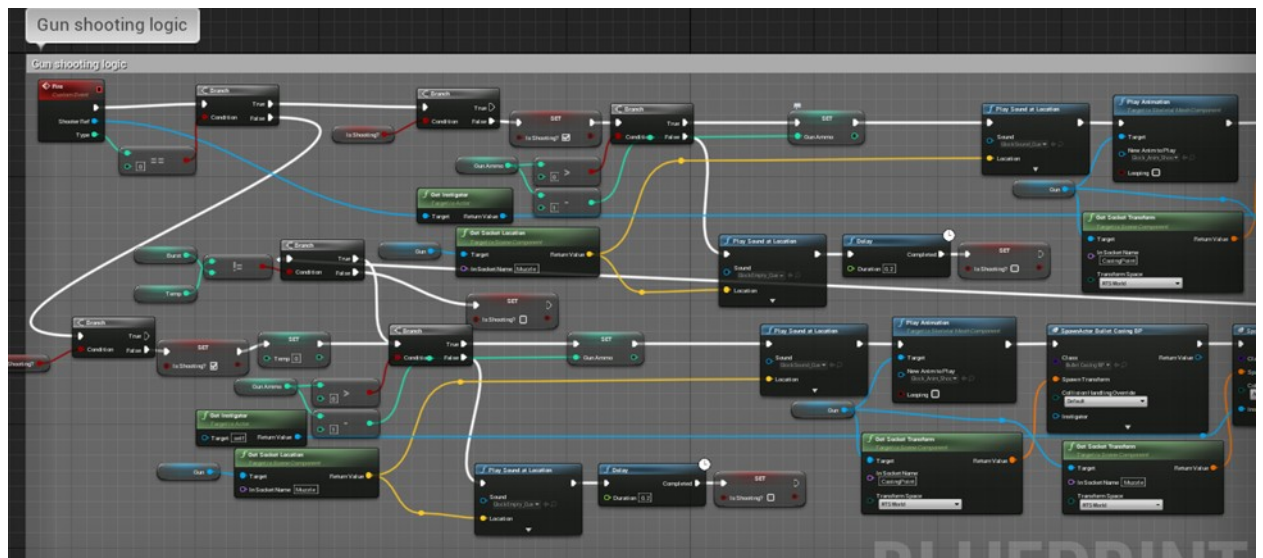


Pav. 2.5 Ginklo blykstės medžiagos grafas.

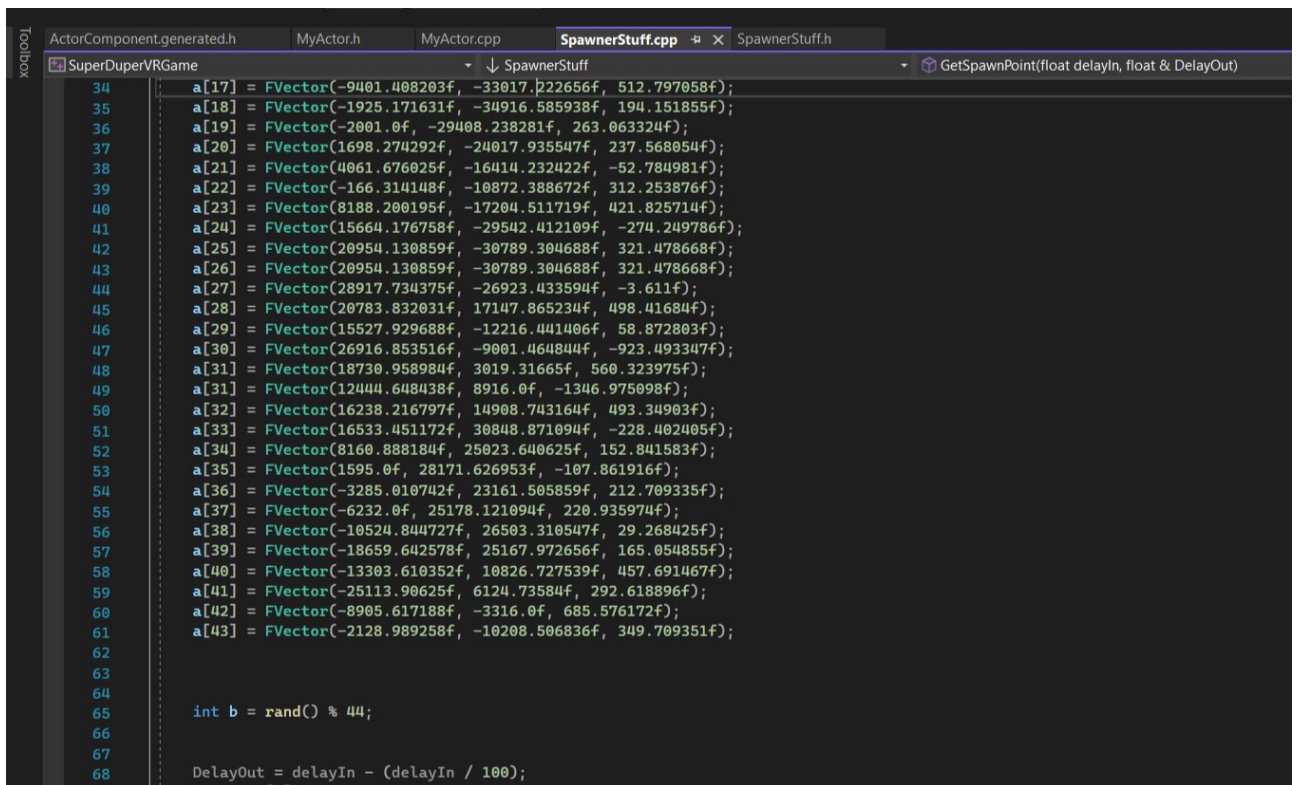
Variklio „Blueprint“ sistema leido greitai sukurti prototipinę žaidimo versiją. Taip pat buvo naudojamos C++ klasės aprašant savo kodą kurį sunku realizuoti Blueprint klasėse.



Pav. 2.6 Pagrindinė žaidimo progreso logikos

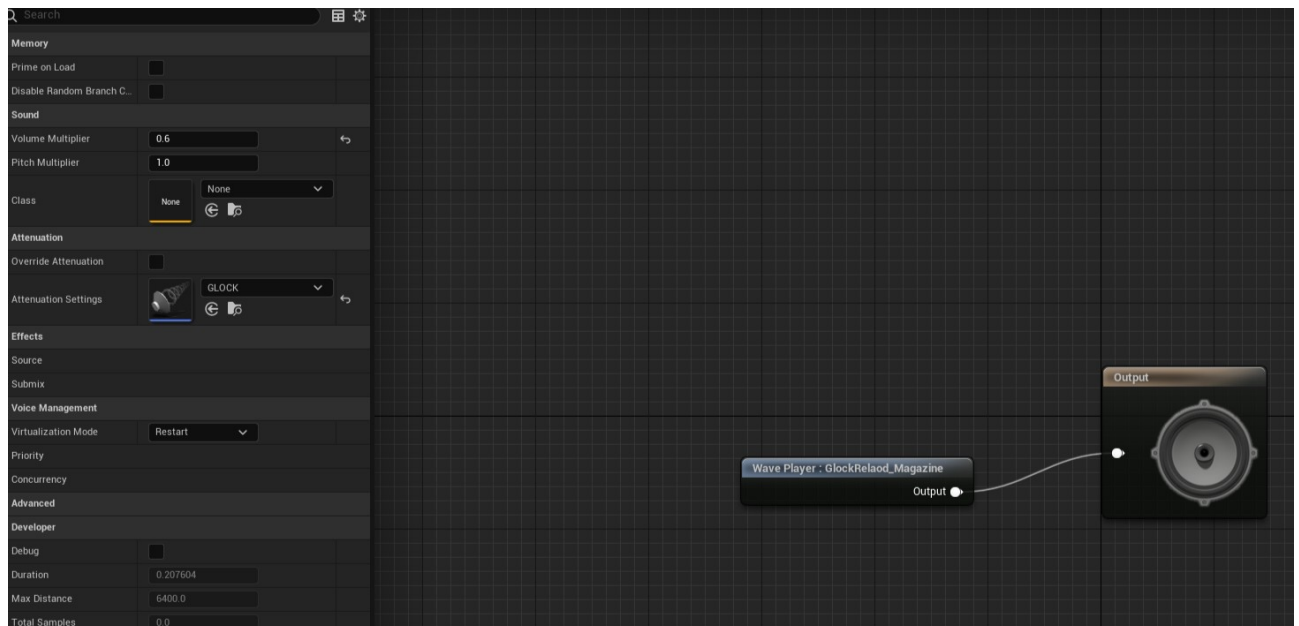


Pav. 2.7 Ginklo šaudymo logikos grafikas



Pav. 2.8 Mūsų pačių kurta C++ klasė

Taip pat buvo išnaudojamas garso redagavimo įrankis. Juo buvo kuriama variacija įvairiems žaidimo garso efektams. Skirtingiems garsams norint pritaikyti skirtingus garso išblėsimo atstumus ir efektus



Pav. 2.9 Garso variacijos grafas.

2.2. Modeliavimas

Modeliavimui naudota „Blender“ programinė įranga. „Blender“ turi modeliavimo, UV redagavimo ir animavimo funkcijas, kurių prireikė kuriant žaidimo modelius ir modelių animacijas. Visi 3D modeliai eksportuoti FBX tipo, nes tai formatas kurį palaiko „Unreal Engine“ žaidimų variklis.

2.3. Versijavimo technologijos

Žaidimo versijavimui buvo naudojama „Git“ sistema ir „GitHub“ talpinimo platforma. Platforma buvo pasirinkta, nes tai yra populiariausia kodo talpinimo platforma. „GitHub“ platformoje patikiamas „Unreal Engine“ projektams skirtas „gitignore“ šabloninis failas leido išvengti nereikalingų ir sugeneruojamų failų versijavimo.

Github repozitorija buvo dalinamasi ir komandos nariai reguliariai atnaujindavo github po žaidimo kodo pakeitimo.

3. Realizacija

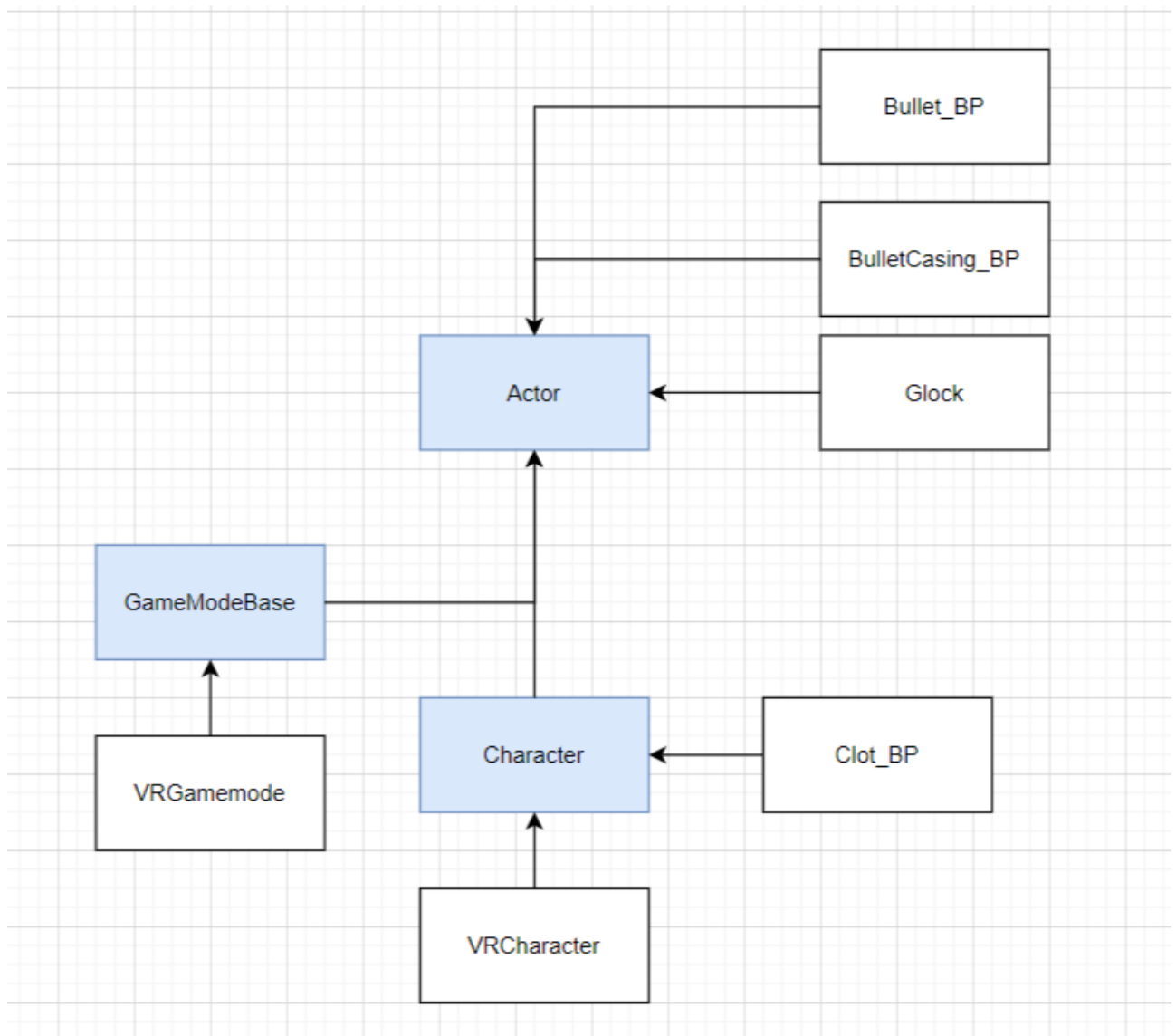
Šiame skyriuje pateikiama informacija apie kiaušinių gaudymo žaidimo realizaciją.

3.1. Projekto architektūra

Aprašomi architektūriniai sprendimai priimti realizuojant programą. Sprendimai priimti norint palengvinti galimus ateities pakeitimus ar praplėtimus.

3.1.1. „Gameplay Framework“

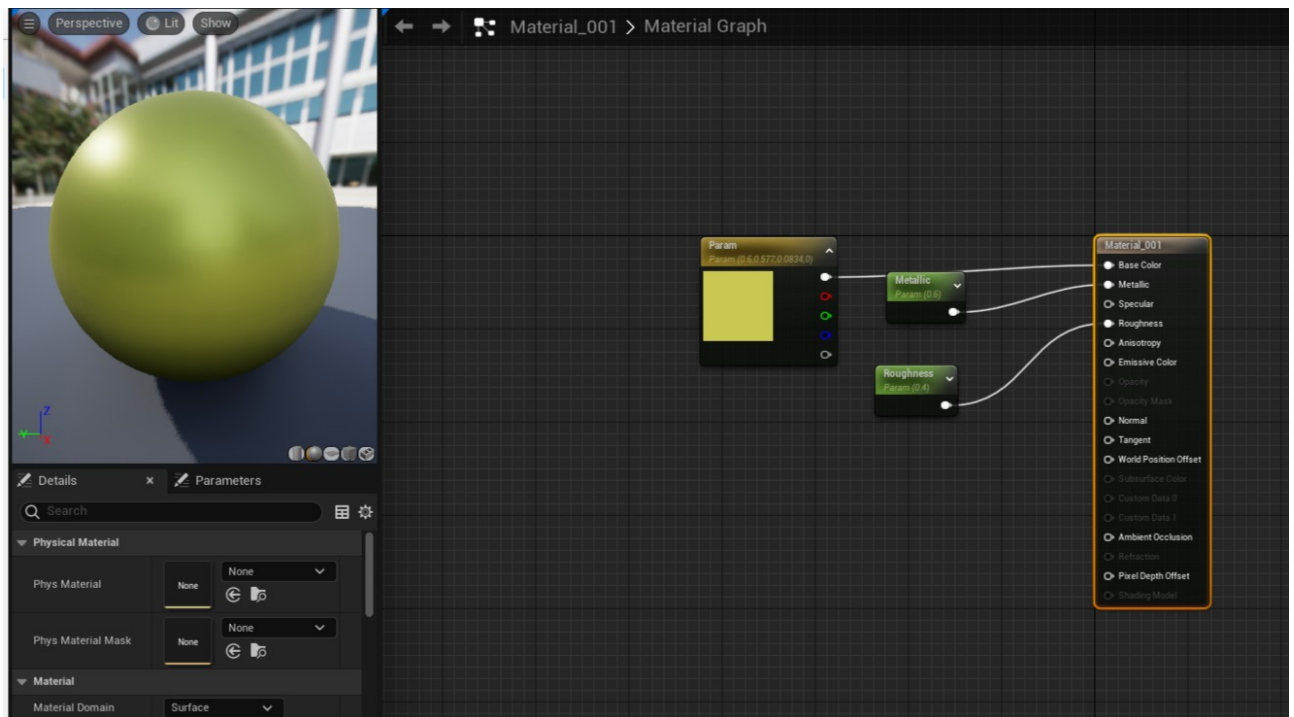
Projektui įgyvendinti buvo naudojama standartinė „Unreal Engine“ „Gameplay Framework“ struktūra. Paveldimos tik trys klasės: „Actor“, „GameModeBase“ ir Character klasės (pav. 3.1). Kitų karkaso klasių paveldėti nereikėjo.



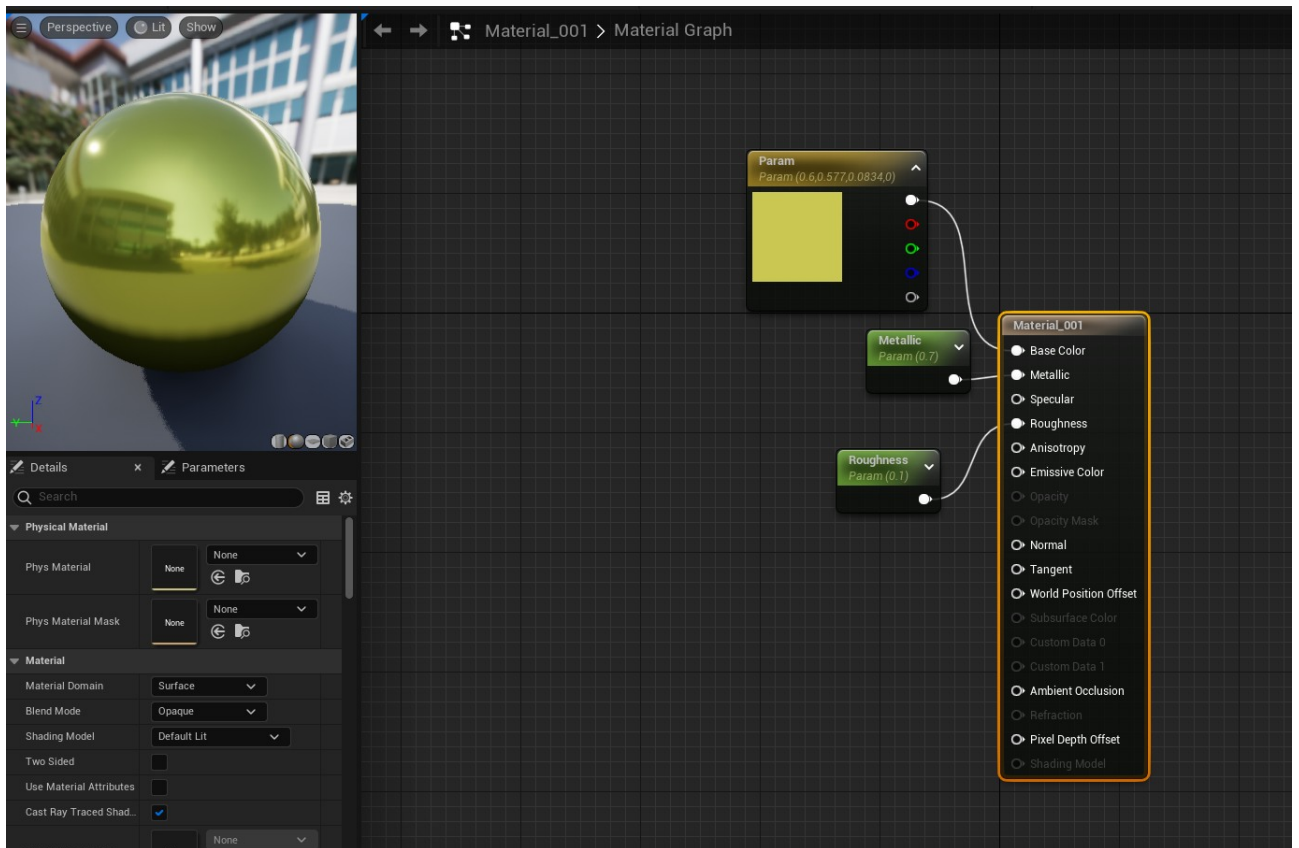
Pav. 3.10 "Gameplay Framework" klasės (mėlynai pažymėtos „Unreal Engine“ klasės)

3.1.2. Medžiagos

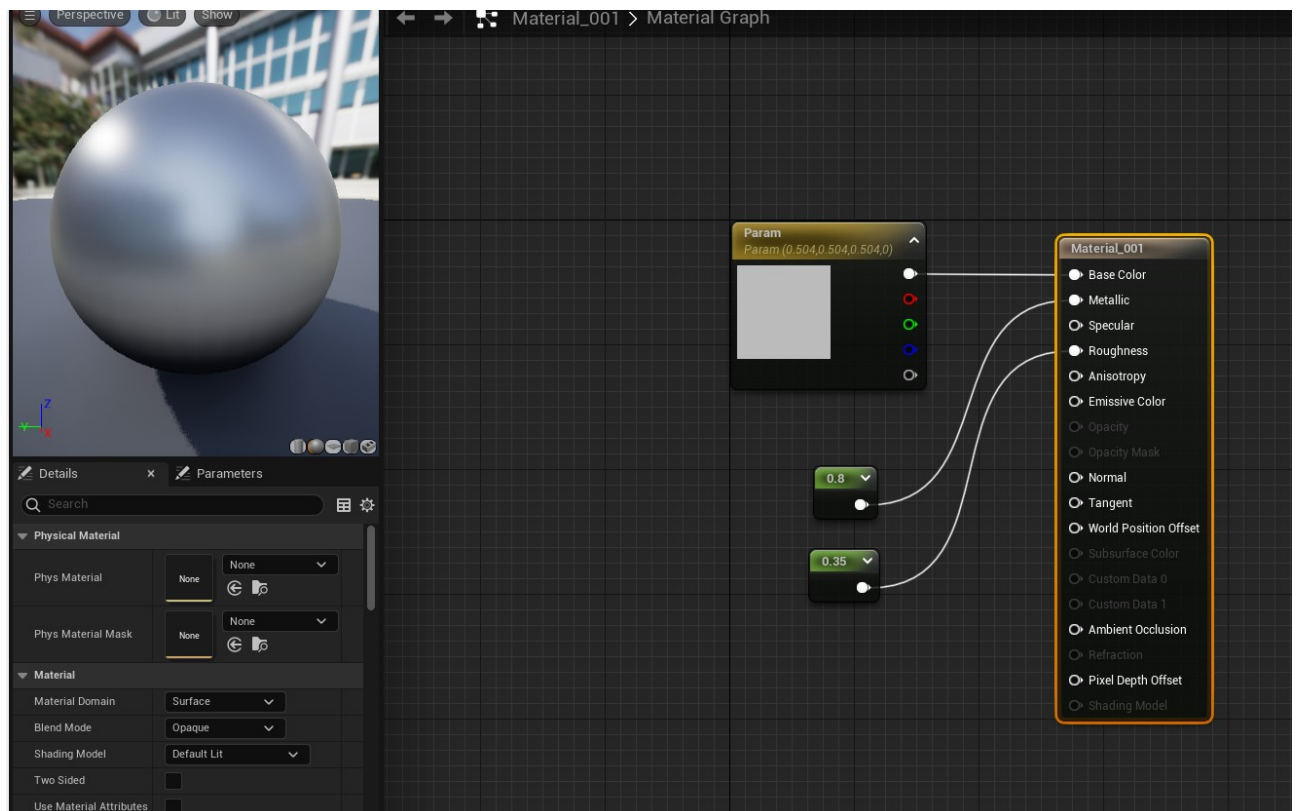
Visiems statiniams modeliams sukurama viena pagrindinė medžiaga (pav. 3.4). Medžiagoje naudojami tekstūrų, vektorių ir skaliariniai parametrai medžiagos būsenai nusakyti. Norint priskirti medžiagą vienam ar kitam objektui yra sukurama pagrindinės medžiagos instancija ir pakeičiami jos parametrai pagal norimą išgauti medžiagą. Tokiu būdu projekte nereikia kompiliuoti daug medžiagų, o norint sukurti daug medžiagų galima naudoti tas pačias tekstūras. Taip pat kuriant modelius naudojantis „Blender“ keletai modelių yra naudojamos po kelias tekstūras išgaunant skirtingų medžiagų efektą.



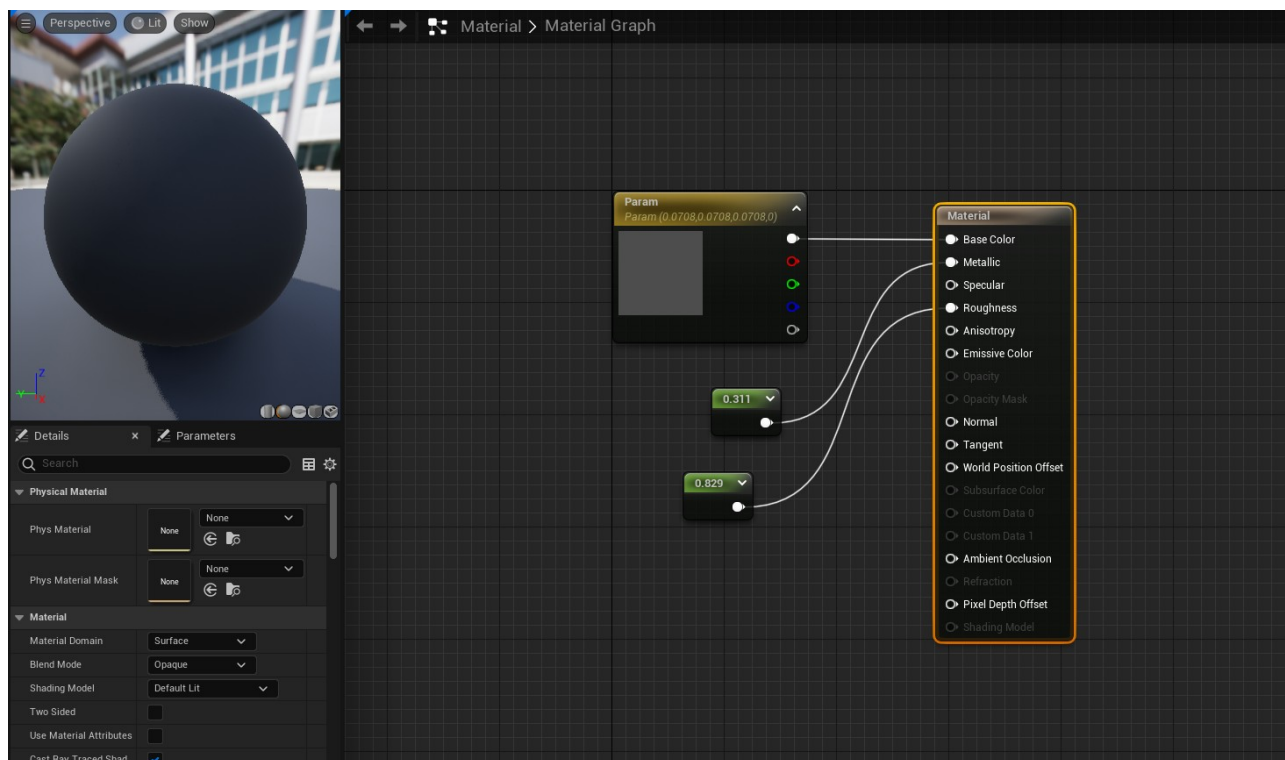
Pav. 3.11 Ginklo kulkos medžiagos grafas



Pav. 3.12 Ginklo gilzės medžiagos grafas

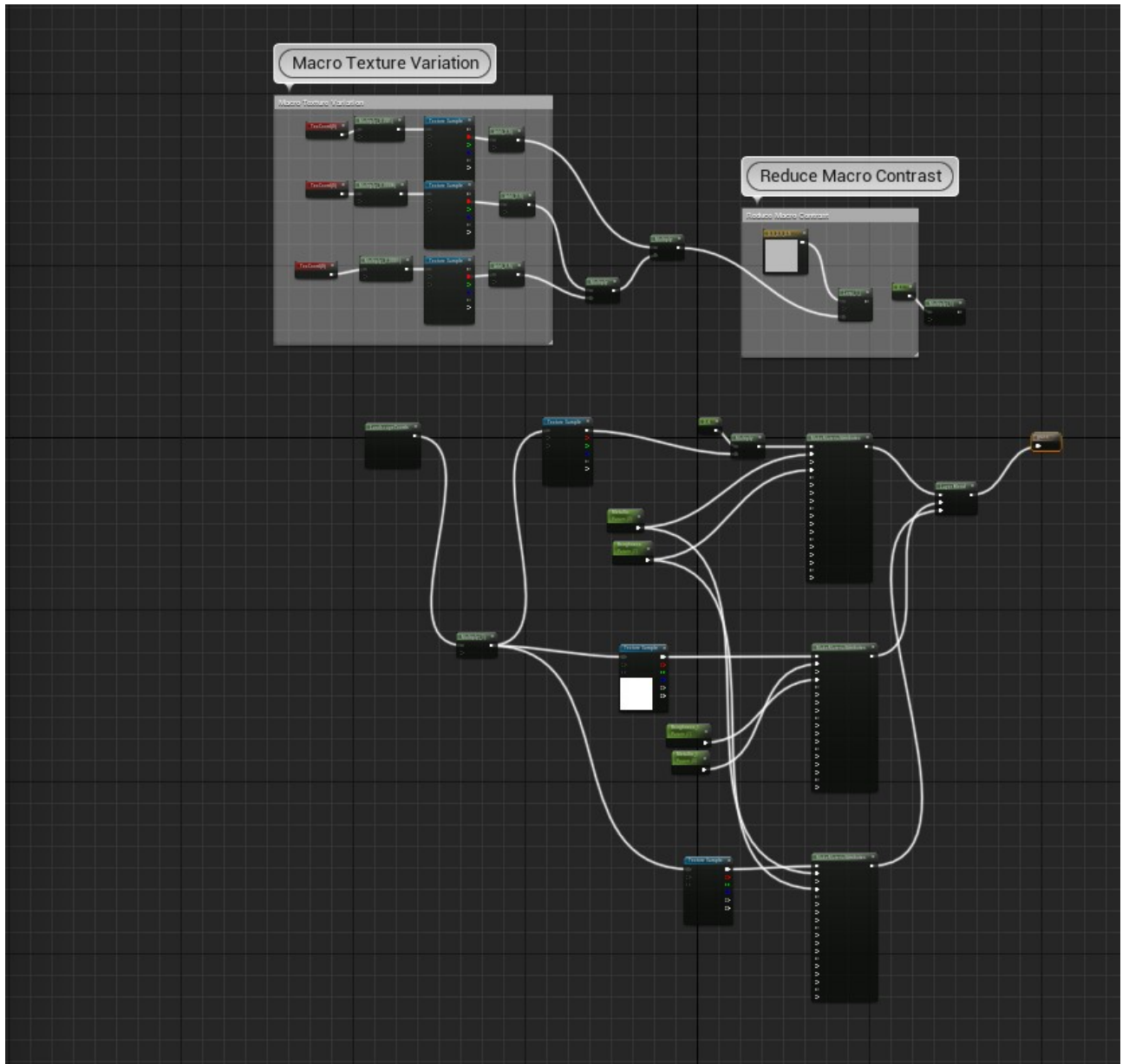


Pav. 3.13 Ginklo rankenos medžiagos grafas

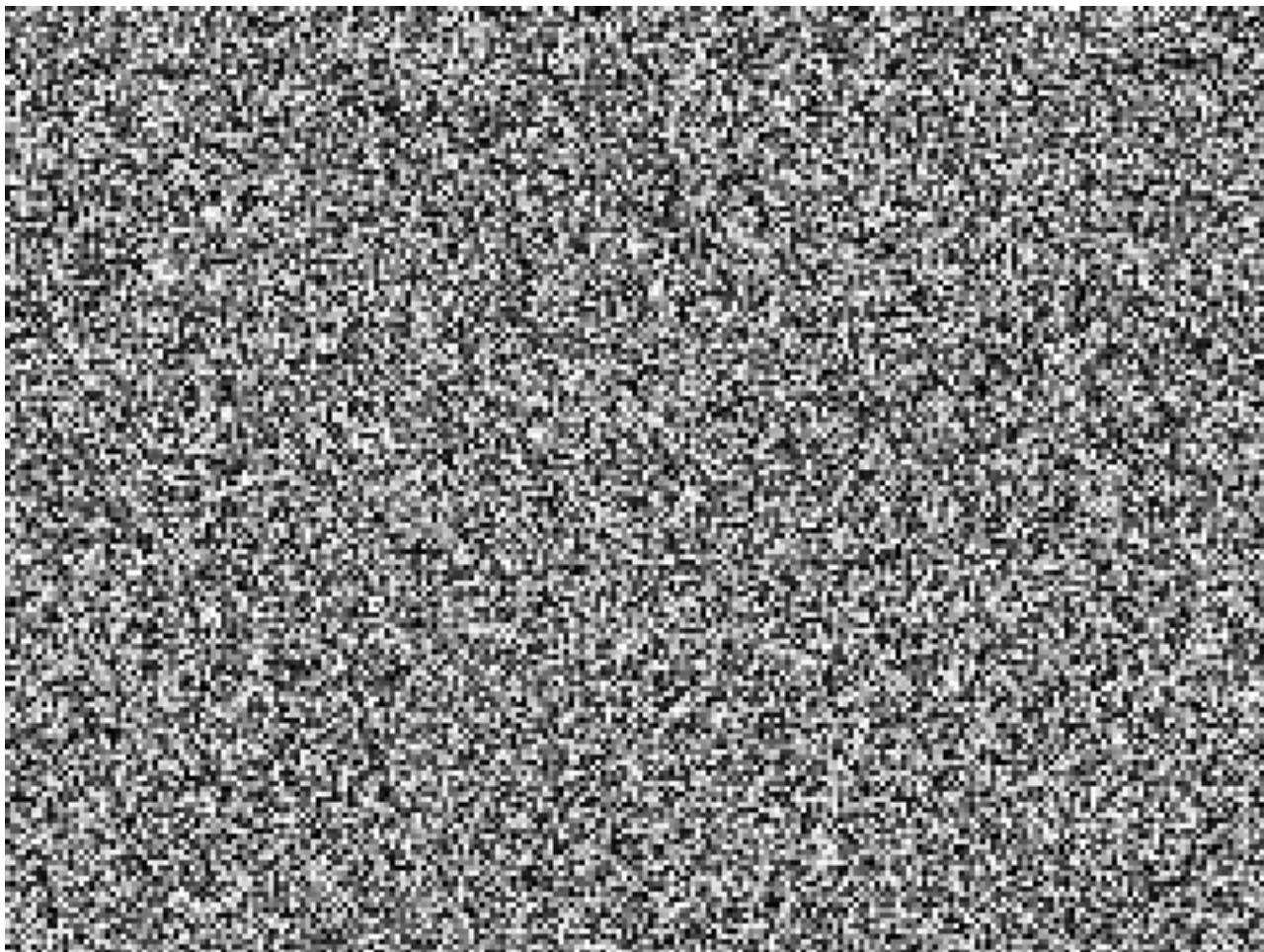


Pav. 3.14 Ginklo viršutinės dalies medžiagos grafas

Tokių tekstūrų kaip žolė plečiasi per visa žaidimo žemėlapi ir vyksta didelis tekstūrų pasikartojimas, todėl sukurtas atsitiktinis medžiagos pasisukimas ir spalvos pasikeitimas naudojantis tekstūrų variacija perlinio triukšmu.



Pav. 3.15 Žolės medžiagos grafas



Pav. 3.16 Perlino garso tekstūra

3.2. Realizacijos detalės

Visą projektą sudaro 1 C++ klasė ir 7 „Blueprint“ klasių. Žaidimui buvo sukurti 9 modeliai iš kurių 2 turi 7 animacijas. Žaidime panaudotas 8 garso failai.

Visi žaidimo garsai ir 3d modeliai yra kurti mūsų pačių, kas leido sukurti vientisą garso ir modelių stilių, leidžiantį žaidėjams labiau isitraukti į žaidimą.

3.3. Realizacijos iššūkiai

Aukšti „Unreal Engine“ reikalavimai sistemai trukdė greitai realizuoti projektą, ypač kai geriausia vazdo plokštė kurią turėjome buvo „gtx 1650 mobile“. Kodo kompiliavimas, redaktorių atidarymas, projekto eksportavimas kiekvieną kartą užtrukdavo. Augant projektui lėtėjo žaidimo „preview“ kadrai per sekundę, kas priverte mus ieškoti žaidimo variklo optimizavimo nustatymuose. Taip pat paskutinėse žaidimo stadijose „Unreal Engine“ pastrigdavo ar išsijungdavo.

Kilo keblumų norint rodyti žaidimo statistikas. Noreta rodyti naudojant HUD klases, tačiau pastebėta kad naudojantis vr iranga „Unreal Engine“ nerodo widget klasių.

Taip pat buvo užsibrėžtas labai didelis žaidimo tikslas, ko nepavyko įgyvendinti, tokiu kaip ginklų parduotuvė tarp kiekvienos priešų bangos ar žaidimo boso paskutinėje bangoje.

Rezultatai ir išvados

1. Žaidimo valdymui panaudoti visi oculus quest 2 valdikliai. Tokia konfigūracija leido žaidėjams labiau įsijausti į žaidimą.
 2. Žaidimo kūrimui pasirinktas „Unreal Engine“ žaidimų kūrimo variklis. Šis variklis integruoja didžiąją dalį VR platformų ir turi reikalingus įrankius norimoms žaidimo funkcijoms sukurti. Itin pravartus buvo medžiagų redaktorius, kurio dėka buvo galima lengvai sukurti specialiuosius efektus žaidimo elementams.
 3. Pasirinkta žaidimo architektūra leidžia paprastai plėsti žaidimo funkcijas
 4. Dėl užsibrėžto per didelio žaidimo tikslo nepavyko įgyvendinti daug planuotų žaidimo funkcijų, kaikurios žaidimo funkcijos turėjo būti supaprastintos arba „scrapintos“.
- Projektas patalpintas šioje repozitorijoje: <https://github.com/Gearsas/SuperDuperVRGame>