

Renderiranje interaktivne računalne grafike u stvarnom vremenu za potrebe razvoja računalne igre u Unreal Engine-u

Lazić, Patrik

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Graphic Arts / Sveučilište u Zagrebu, Grafički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:216:994134>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-11**



Repository / Repozitorij:

[Faculty of Graphic Arts Repository](#)



SVEUČILIŠTE U ZAGREBU

GRAFIČKI FAKULTET

PATRIK LAZIĆ

RENDERIRANJE INTERAKTIVNE RAČUNALNE
GRAFIKE U STVARNOM VREMENU ZA POTREBE
RAZVOJA RAČUNALNE IGRE U UNREAL ENGINE-U

DIPLOMSKI RAD

Zagreb, 2023.



Sveučilište u Zagrebu
Grafčki fakultet

PATRIK LAZIĆ

RENDERIRANJE INTERAKTIVNE RAČUNALNE
GRAFIKE U STVARNOM VREMENU ZA POTREBE
RAZVOJA RAČUNALNE IGRE U UNREAL ENGINE-U

DIPLOMSKI RAD

Mentor:

izv.prof.dr.sc. Tibor Skala

Student:

Patrik Lazić

Zagreb, 2023.

SVEUČILIŠTE U ZAGREBU

GRAFIČKI FAKULTET

Getaldićeva 2

Zagreb, 15. 9. 2023.

Temeljem podnijetog zahtjeva za prijavu teme diplomskog rada izdaje se

RJEŠENJE

kojim se studentu/ici Patriku Laziću, JMBAG 0268015176, sukladno čl. 5. st. 5. Pravilnika o izradi i obrani diplomskog rada od 13.02.2012. godine, odobrava izrada diplomskog rada, pod naslovom: Renderiranje interaktivne računalne grafike u stvarnom vremenu za potrebe razvoja računalne igre u Unreal Engine-u, pod mentorstvom izv. prof. dr. sc. Tibora Skale.

Sukladno čl. 9. st. 1. Pravilnika o izradi i obrani diplomskog rada od 13.02.2012. godine, Povjerenstvo za nastavu, završne i diplomske ispite predložilo je ispitno Povjerenstvo kako slijedi:

1. doc. dr. sc. Rudolf Maja, predsjednik/ica
2. izv. prof. dr. sc. Skala Tibor, mentor/ica
3. doc. dr. sc. Stanić Loknar Nikolina, član/ica



SAŽETAK:

U ovom diplomskom radu prikazan je proces izrade video igre u Unreal Engine-u 5 s naglaskom na razvoj igre i korištenje naprednih tehnologija softvera. Igra koja je razvijena je distopijska pucačina iz prvog lica (eng. First-person shooter, FPS) s elementima preživljavanja, a koristit će se napredne tehnologije poput Nanite-a i Lumena kako bi se postigla željena kvaliteta vizualnog rješenja.

Kroz rad objašnjene su ključne značajke Unreal Engine-a 5, poput njegovih mogućnosti renderiranja u realnom vremenu, dinamične globalne osvjetljenosti i simulacije fizike. Također se razrađuje korištenje različitih ugrađenih mogućnosti engine-a za razvoj igara, uključujući alate za 3D modeliranje i alate za animaciju.

Kako bi se postigla željena kvaliteta vizualne izvedbe, koriste se napredne tehnologije poput Nanite tehnologije za brzo učitavanje visokokvalitetnih modela u igru, bez obzira na veličinu i složenost, kao i Lumen tehnologije za postizanje realističnog osvjetljenja i sjena u igri. Korištenje ovih tehnologija omogućuje da svaki element igre oživi, stvarajući dojam stvarnog, imerzivnog iskustva.

Glavni dio rada opisuje izradu različitih elemenata igre. Kreiranje objekata i tekstura koje su inspirirane distopijskom estetikom, animacija funkcija igrača u prvome licu, te sam dizajn interaktivnog prostora nivoa unutar igre.

Rezultat rada je finalno rješenje u obliku video igre koja predstavlja sveobuhvatan proces razvoja igre korištenjem Unreal Engine-a prikazujući njegovu snagu za stvaranje imerzivnog i uzbudljivog iskustva.

Ključne riječi: Renderiranje u realnom vremenu, interaktivna računalna grafika, Unreal engine, razvoj računalne igre

SUMMARY:

This thesis presents the process of creating video games in Unreal Engine 5 with an emphasis on game development and the use of advanced technological software. The game that has been developed is a dystopian first-person shooter (FPS) with survival elements, and advanced technologies such as Nanite and Lumen will be used to achieve the desired quality of the visual solution.

The paper explains key features of Unreal Engine 5, such as its real-time rendering capabilities, dynamic global lighting, and physics simulation. The use of the engine's various built-in game development capabilities, including 3D modeling and animation tools, is also being developed.

In order to achieve the desired quality of visual performance, advanced technologies such as Nanite technology are used to quickly load high-quality models into the game, regardless of size and complexity, as well as Lumen technology to achieve realistic lighting and shadows in the game. The use of these technologies allows every element of the game to come to life, creating the impression of a real, immersive experience.

The main part of the paper describes the creation of different elements of the game. The creation of objects and textures inspired by dystopian aesthetics, the animation of player functions in the first person, and the design of the interactive space of the levels within the game.

The result of the work is the final solution in the form of a video game that represents a comprehensive game development process using the Unreal Engine that showcases its power to create an immersive and exciting experience.

Keywords: Real-time rendering, interactive computer graphics, Unreal engine, computer game development

SADRŽAJ:

1. UVOD	1
2. TEORIJSKI DIO	2
2.1 GAME ENGINE.....	2
2.2 UNREAL ENGINE	3
2.3 RENDERIRANJE	4
2.3.1 SCANLINE.....	5
2.3.2 RAYTRACING.....	6
2.3.3 RADIOSITY.....	8
2.4 RENDERIRANJE U STVARNOM VREMENU.....	9
2.5 BLUEPRINTOVI.....	12
2.6 VARIABLE.....	13
2.7 SUSTAV KOLIZIJE I FIZIKE.....	14
3. PRAKTIČNI DIO	15
3.1 PEJZAŽNI SUSTAV	15
3.2 FOLIAGE.....	18
3.3 AUTO MATERIJAL.....	20
3.4 TEHNIKE OSVIJETLJENA I GEOMETRIJE	28
3.4.1 NANITE.....	28
3.4.2 DINAMIČNA GLOBALNA OSVIJETLJENOST.....	31
3.4.3 LUMEN.....	32
3.5 ANIMACIJA IGRAČA.....	34
3.6 FINALNI REZULTAT.....	40
3.7 METODOLOGIJA ISTRAŽIVANJA.....	46
3.4. ANALIZA I REZULTATI ANKETE.....	49
3.5. RASPRAVA REZULTATA	62
4. ZAKLJUČAK	65
5. LITERATURA.....	66
6. POPIS SLIKA	69
7. POPIS MANJE POZNATIH RIJEČI	72

1. UVOD

Grafički alati za izradu interaktivne računalne grafike danas su moćna razvojna okruženja koja iz temelja mijenjaju način na koji se stvara vizualno impresivan digitalni sadržaj. Za potrebe pisanja ovog rada korišten je Unreal Engine kao alat za izradu računalne grafike. Jedna od velikih vrijednosti ovoga game engine-a je njegova podrška za renderiranje u stvarnom vremenu. Riječ je o tehnici trenutnog iscrtavanja slike koja programerima i dizajnerima omogućuje da izvrše promjene na sceni u stvarnom vremenu, čime se cijeli postupak izrade grafike značajno ubrzava uz brže iteracije i povratne informacije. To znači da programeri i razvojni inženjeri mogu odmah vidjeti utjecaj promjena na svijet igre, osvjetljenje ili efekte, bez potrebe za dugotrajnim renderiranjem. „Engine“ za dizajn računalnih igara je softversko okruženje za razvoj koji se naziva „arhitektura igre“, „framework igre“ i „jezgra igre“ s postavkama i konfiguracijama koje optimiziraju i pojednostavljaju razvoj videoigara na različitim programskim jezicima. Alat uključuje mehanizam za renderiranje 2D ili 3D grafike koja je kompatibilna s različitim import formatima, mehanizam fizike koji simulira aktivnosti u stvarnom svijetu, umjetnu inteligenciju (eng. artificial intelligence - AI) koja automatski reagira na radnje igrača, mehanizam zvuka koji kontrolira zvučne efekte, te mehanizam za animaciju i niz drugih značajki.

Kroz ovaj rad u teoretskom dijelu bit će opisane funkcije i mogućnosti game engine-a kao cjeline, te samog Unreal Engine-a, kao jednog od najmoćnijih razvojnih okruženja za izradu interaktivne računalne grafike. U sklopu praktičnog dijela u kojem je razvijena računalna igra prikazat će se vizualni sadržaj dobiven manipulacijom „blueprint“ sistema Unreal engine-a u jednom od trenutno najpopularnijih tehnika renderiranja.

2. TEORIJSKI DIO

2.1 GAME ENGINE

Izraz „game engine“ pojavio se sredinom 1990-ih kao referenca na vrlo popularne igre u prvom licu „Doom“ od kompanije pod imenom „id Software“ i „Unreal“ od Epic Games-a. Doom je projektiran s prilično dobro definiranim odvajanjem između svoje primarne softverske komponente (kao što je sustav za prikaz trodimenzionalne grafike, sustav za otkrivanje kolizija ili audio sustav) i umjetničkih sredstava, svijeta igre i pravila igre koji su činili igračevo cjelokupno iskustvo interakcije. Vrijednost ovog odvajanja postala je očita kada su programeri počeli licencirati igre i njihovo preoblikovanje u nove proizvode stvaranjem novog dizajna, „layouta“ svijeta, oružja, likova, vozila i pravila igre sa samo minimalnim promjenama u game engine-u. Ovo je označilo rođenje „mod zajednice“— skupina pojedinačnih igrača i malih neovisnih studija koji su izgradili nove igre mijenjanjem postojećih igara, korištenjem besplatnih alata koji pružaju izvorni programeri. Pred kraj 1990-ih, neke igre poput Quake III Arene i Unreal-a dizajnirani su imajući na umu ponovnu upotrebu i „moding“. Game engine-i su postali visoko prilagodljivi putem skriptnih jezika kao što je ID-ov Quake C. Licenciranje game engine-a postao je održiv sekundarni izvor prihoda za programere koji su ih kreirali. Danas programeri igara mogu kupiti, ili dobiti licencu game engine-a i ponovno koristiti značajne dijelove svojih ključnih softverskih komponenti za izradu igara. Dok ova praksa još uvijek uključuje znatna ulaganja, to može biti puno ekonomičnije od razvoja cjelokupnog game engine-a za svaku kompaniju zasebno. [1]

Game engine-i obično su donekle specifični za žanr. Softver dizajniran za borbene igre dvije osobe u boksačkom ringu bit će poprilično drukčiji od onoga za masovne online igre s više igrača (eng. Massively multiplayer online game, MMOG) ili pucačine iz prvog lica (eng. First-person shooter, FPS) ili strategije u stvarnom vremenu (Real-time strategy, RTS). Međutim, danas postoje i velika preklapanja—sve 3D igre, bez obzira na žanr, zahtijevaju neki oblik korisničkog unosa, 3D mrežu prikazivanja uključujući prikazivanje teksta u mnoštvo fontova, kvalitetan audio sustav i slično.

Unreal Engine, na primjer, dizajniran je za pucačine u prvom licu, ali uspješno se koristi i za kreiranje igara u nizu drugih žanrova, uključujući iznimno popularnu franšizu pucačina iz trećeg lica „Gears of War“ od „Epic Games-a“, akcijsko-avanturističke igre „Batman: Arkham“ serija i druge. [1]

2.2 UNREAL ENGINE

Unreal Engine je game „engine“ razvijen od strane kompanije „Epic Games“. On predstavlja sveobuhvatno softversko rješenje koje omogućava razvoj visokokvalitetnih računalnih igara, virtualne stvarnosti, proširene stvarnosti i simulacija.

Unreal Engine radi na principu renderiranja u stvarnom vremenu (eng. real-time renderiranja), što znači da se slike i animacije generiraju i prikazuju u stvarnom vremenu dok se igrač kreće ili interagira s okruženjem igre. Ova tehnologija omogućava brz i fluidan pristup dizajnu igre.

Ključni element Unreal Engine-a je njegov editor, poznat kao „Unreal Editor“, koji je moćan vizualni alat za kreiranje svijeta igre. U ovom editoru, programeri i umjetnici mogu kreirati i modificirati 3D modele, teksture, materijale, osvjetljenje, animacije i druge elemente igre. Editor koristi intuitivno sučelje koje omogućava jednostavno organiziranje i uređivanje igračkog sadržaja.

Unreal Engine ima niz editora koji pomažu u kreiranju igre. Na slici 1. prikazan je „Unreal Editor“, početni prozor za Unreal Engine. Može se smatrati kao glavni uređivač koji omogućuje pristup drugim podsustavima, kao što su „Blueprint“ i „Material“ sistemi. Unreal Editor pruža vizualno sučelje sastavljeno od okvira za prikaz i prozora koji omogućuju uvoz, organiziranje, uređivanje i dodavanje ponašanja u interakcijama „aseta“ . Drugi sub-editori imaju vrlo specijalizirane funkcije koje omogućuju kontrolu detalja „aseta“ (kako izgleda, kako se ponaša). [2]



Slika 1: Početni prozor Unreal editora

Izvor: Autor

2.3 RENDERIRANJE

Renderiranje je proces, pomoću softverskog sustava, stvaranja slike iz predloška u posebno definiranim informacijskim strukturama kao što su modeli. Proces renderiranja temeljen je na podacima o geometriji, gledištu, teksturi, osvjetljenju i sjenčanju koji opisuju virtualnu sliku. [3]

Renderiranje obuhvaća simulaciju interakcije svjetlosti s objektima u virtualnoj sceni kako bi se generirala fotorealistična slika te ima ključnu ulogu u stvaranju vizualnih iskustava u širokom spektru primjena, uključujući filmsku industriju, računalne igre, medicinsku vizualizaciju te arhitektonsko i produktno projektiranje. [3]

Postoje dvije kategorije renderiranja: renderiranje unaprijed (eng. Pre-rendering) i renderiranje u stvarnom vremenu (eng. Real-time rendering).

Renderiranje u stvarnom vremenu: Istaknuta tehnika renderiranja koja se koristi u interaktivnoj grafici i igricama gdje se slike moraju stvarati velikom brzinom. Budući da je interakcija korisnika visoka u takvim okruženjima, potrebno je stvaranje slike u stvarnom vremenu. Namjenski grafički hardver i prethodno kompiliranje dostupnih

informacija poboljšali su i omogućili današnje performanse renderiranja u stvarnom vremenu.

Renderiranje unaprijed: Tehnika renderiranja koja se koristi u okruženjima u kojima brzina nije važna, a izračuni slike izvode se pomoću više jezgrenih procesorskih jedinica, a ne grafičkog hardvera. Ova tehnika renderiranja se uglavnom koristi u animaciji i vizualnim efektima, gdje fotorealizam treba biti na najvišem mogućem standardu.

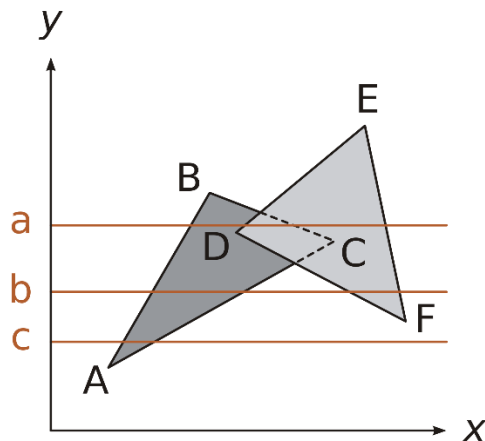
Za ove vrste renderiranja koriste se tri glavne računalne tehnike:

- Scanline
- Raytracing
- Radiosity

2.3.1 SCANLINE

„Scanline“ algoritam temelji se na ideji obrade slika liniju po liniju, „scanline“ po „scanline“. Ovaj pristup omogućuje učinkovitu i sistematičnu obradu površina i objekata u sceni. Na slici 2. prikazan je „scanline“ algoritam gdje svi prikazani poligoni prvo se sortiraju prema gornjoj y-koordinati na kojoj se prvi put pojavljuju. Zatim za svaki redak skenirana linija slike izračunava njegovo sjecište s početka sortiranog popisa. Tijekom postupka, sortirani popis se kontinuirano ažurira kako bi se uklonili poligoni koji postanu nevidljivi dok se aktivna linija skeniranja kreće niz sliku. Svaki „scanline“ predstavlja horizontalnu liniju na rasterskom zaslonu, a algoritam se koristi za izračunavanje i primjenu svjetlosnih efekata na objekte koji se protežu preko tog „scanline-a“. [3]

„Scanline“ algoritam ima nekoliko ključnih koraka. Prvo, detektira se kada objekt prelazi ili sječe „scanline“. Zatim se izračunava točka presjecišta između objekta i „scanline-a“. Na temelju tih podataka, primjenjuju se različite tehnike shadeliranja, teksturiranja, te izračunavaju sjene i osvjetljenje. [3]



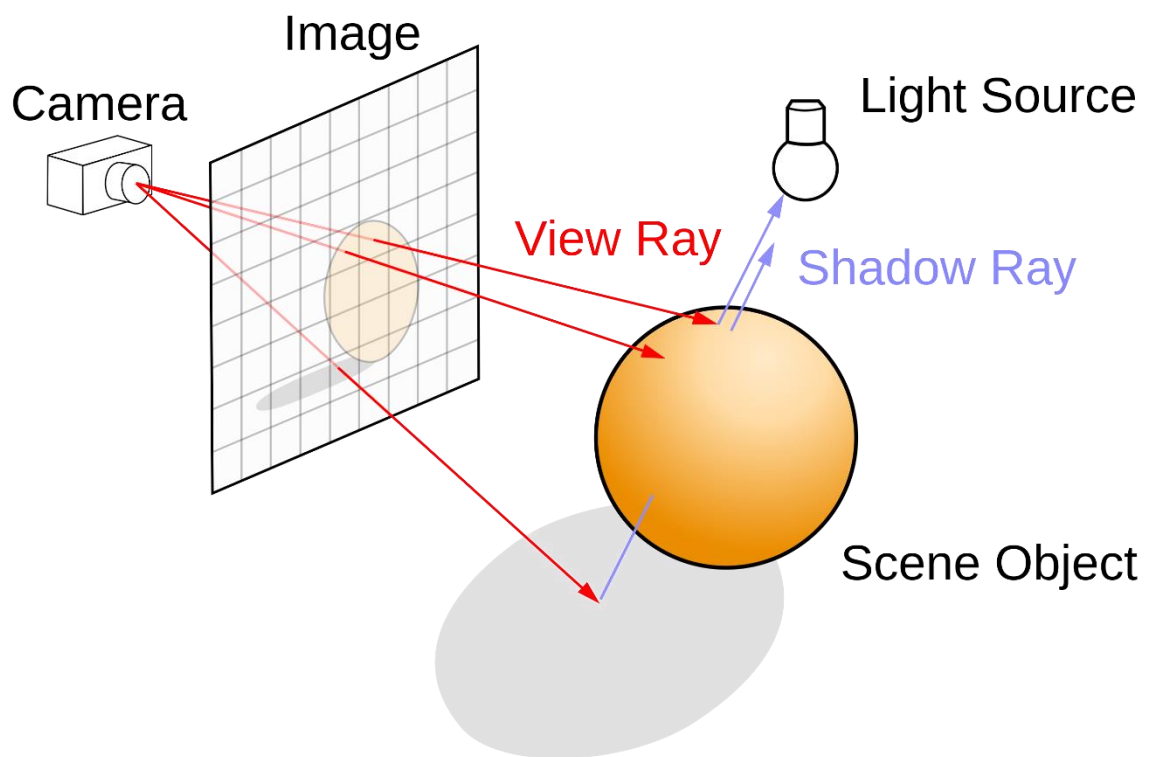
Slika 2: Scanline algoritam

Izvor: https://en.wikipedia.org/wiki/Scanline_rendering#/media/File:Scanline_algorithm.svg

„Scanline“ algoritam se često koristi u procesu rasterizacije, pristupu renderiranju koji konvertira trodimenzionalne podatke u dvodimenzionalne slike. Ovaj pristup omogućava brzo renderiranje u stvarnom vremenu, čineći ga ključnim za interaktivne aplikacije poput računalnih igara. Njegova efikasnost proizlazi iz činjenice da se svaki „scanline“ obrađuje neovisno, što omogućava paralelnu obradu i brže renderiranje. Također, „scanline“ algoritam je posebno prikladan za obradu konveksnih objekata i površina. Međutim, algoritam se može suočiti s ograničenjima kod obrade konkavnih površina i objekata te u situacijama gdje objekti djelomično prekrivaju jedan drugoga. [3]

2.3.2 RAYTRACING

„Raytracing“ se temelji na praćenju putanja svjetlosnih zraka kroz scenu kako bi se simulirale različite svjetlosne interakcije. Svaka svjetlosna zraka (eng. Ray) emitira se iz kamere kroz svaki piksel ekrana, prema sceni. „Raytracing“ precizno modelira refleksije, refrakcije, sjene, osvjetljenje i difuzne refleksije svjetlosti na površinama objekata. [3]



Slika 3: Građenje slike sa „Ray-tracing“ algoritmom

Izvor:

[https://en.wikipedia.org/wiki/Ray_tracing_\(graphics\)#/media/File:Ray_trace_diagram.svg](https://en.wikipedia.org/wiki/Ray_tracing_(graphics)#/media/File:Ray_trace_diagram.svg)

Na slici 3. prikazano je kako „Ray-tracing“ prolazi kroz niz ključnih koraka. Za svaki piksel ekrana, prati se primarna zraka od kamere prema sceni kako bi se pronašla točka presjecišta s objektima. Zatim se primjenjuju refleksije i refrakcije putem rekurzivnih zraka. Dodatno, se prate zrake sijena (eng. Shadow rays) kako bi se odredile sjene i sekundarne zrake (eng. secondary rays) kako bi se modelirale složenije svjetlosne interakcije. [3]

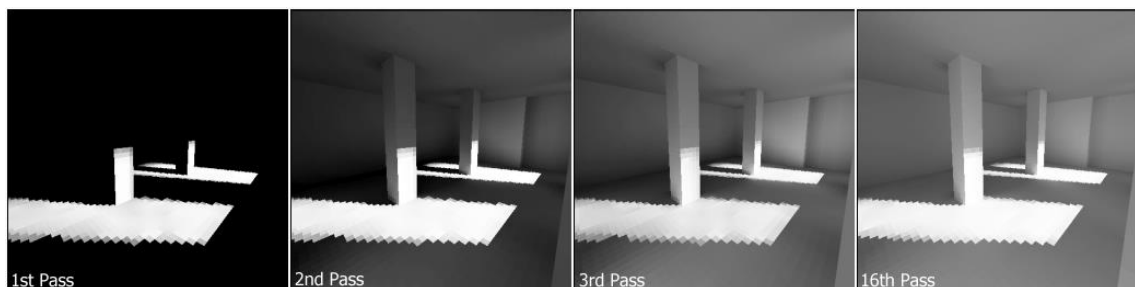
„Ray-tracing“ je poznat po svojoj sposobnosti postizanja visokog stupnja fotorealizma. Precizna simulacija svjetlosnih tehnika, kao što su meke sjene, globalna iluminacija i suptilni svjetlosni efekti, omogućuje generiranje slika koje su izuzetno vjerne stvarnom svijetu. Ovaj pristup često se koristi u filmskoj produkciji i arhitektonskoj vizualizaciji gdje je kvaliteta slike od presudne važnosti.

„Ray-tracing“ je zahtjevan proces koji zahtijeva značajne računalne resurse, posebno za kompleksne scene. Vrijeme renderiranja može biti produženo zbog rekurzivnih zraka i složenih svjetlosnih simulacija.

2.3.3 RADIOSITY

„Radiosity“ se temelji na ideji razmjene svjetlosne energije između površina objekata u sceni. Ovaj pristup modelira kako površine reflektiraju svjetlost na druge površine, uzimajući u obzir difuzno i neizravno osvjetljenje. „Radiosity“ pruža realističan prikaz osvjetljenja i globalne iluminacije, posebno za scene s nehomogenim materijalima i kompleksnim geometrijama.

Na slici 4. prikazan je proces koji se sastoji od nekoliko ključnih koraka. Prvo, scena se razbija na diskretne površinske elemente, tzv. dijelove (eng. Patcheve). Zatim se izračunava faktor prijenosa svjetlosne energije između dijelova, uzimajući u obzir reflektirane svjetlosne zrake. Ovaj proces se ponavlja iterativno dok se ne postigne konvergencija, odnosno stabilno stanje ravnoteže svjetlosne energije. [4]



Slika 4: Koraci procesa „Radiosity“

Izvor:

[https://en.wikipedia.org/wiki/Radiosity_\(computer_graphics\)#/media/File:Radiosity_Progress.png](https://en.wikipedia.org/wiki/Radiosity_(computer_graphics)#/media/File:Radiosity_Progress.png)

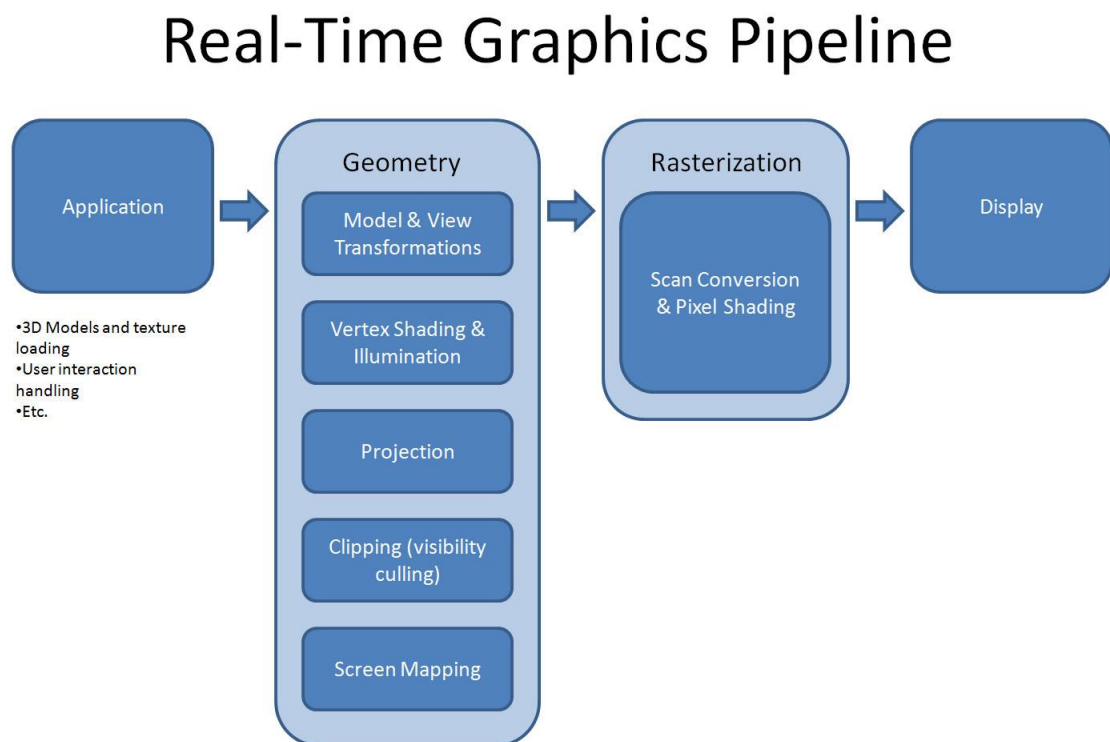
„Radiosity“ se posebno koristi u fotorealističnoj vizualizaciji, gdje se visok stupanj osvjetljenja i realističnosti cijene. Ovaj pristup je neizostavan u arhitektonskim vizualizacijama, simulacijama osvjetljenja unutarnjih prostora, te u filmskoj industriji za postizanje autentičnih svjetlosnih efekata. Takva metoda omogućava postizanje visokog stupnja realizma u osvjetljenju i reprodukciji mekih sjena i neizravnog osvjetljenja. Međutim, proces izračuna radiosity-a može biti računalno zahtjevan i vremenski dugotrajan, posebno za kompleksne scene s velikim brojem površina. [4]

2.4 RENDERIRANJE U STVARNOM VREMENU

Renderiranje u stvarnom vremenu se odnosi na brzu generaciju slika na računalu i predstavlja najdinamičnije područje računalne grafike. Slike se trenutačno prikazuju na zaslonu, dok gledatelj aktivno sudjeluje ili reagira na događaje, a ta uzajamna interakcija ima utjecaj na ono što će se sljedeće stvoriti. Ovaj proces reakcije i prikaza odvija se brzinom koja je dovoljno visoka da gledatelj ne percipira pojedinačne slike. Brzina prikazivanja slika mjeri se u sličicama u sekundi (eng. Frame per second, FPS) ili Hertzima (Hz). Kada je brzina prikaza samo jedna sličica u sekundi, doživljaj interaktivnosti je minimalan, jer korisnik jasno primjećuje svaku promjenu slike. Kako se brzina prikaza povećava osjećaj interaktivnosti postaje izraženiji. Video igre obično teže postizanju brzina od 60, 120, 144, ili više FPS-a, što dodatno poboljšava fokus korisnika na radnju i poboljšava njihove reakcije. [5]

Ključni aspekt renderiranja u stvarnom vremenu je grafički renderni tok (eng. Pipeline), često nazivan jednostavno „pipeline“. Osnovna svrha ovog toka je generiranje ili renderiranje dvodimenzionalnih slika s virtualnom kamerom, trodimenzionalnim objektima, izvorima svjetla i drugim elementima. Stoga se tok za renderiranje smatra temeljnim alatom za izvođenje renderiranja u stvarnom vremenu. Cilj je precizno definirati položaj i oblik objekata putem njihove geometrije, karakteristika okoline i postavljanja kamere unutar tog okruženja. Izgled objekata nakon toga dalje se oblikuje putem svojstava materijala, izvora svjetla, primijenjenih tekstura i primijenjenih jednadžbi za sjene. [6]

Na slici 5. prikazane su osnovne strukture toka za renderiranje koje se sastoje od četiri ključne faze: aplikacije, obrade geometrije, rasterizacije i obrade piksela. Svaka od ovih faza može funkcionirati kao samostalni tok, što je prikazano u fazi obrade geometrije, ili se faze mogu izvoditi (djelomično) paralelno. Na ovom dijagramu, faza primjene predstavlja jedan proces, iako bi i taj korak također mogao biti izveden kao tok ili paralelno. Važno je napomenuti da rasterizacija određuje piksele unutar osnovne grafičke jedinice, na primjer, unutar oblika kvadrata.



Slika 5: Četiri faze toka renderiranja u stvarnom vremenu

Izvor: <https://www.cgchannel.com/2010/11/cg-science-for-artists-part-2-the-real-time-rendering-pipeline/>

Faza primjene kontrolirana je aplikacijom i obično se implementira u softveru koji se izvodi na centralnim procesorskim jedinicama (eng. Central processing unit, CPU). Ovi procesori često sadrže više jezgri koje omogućuju obradu više niti paralelno. To omogućuje procesorima da učinkovito izvršavaju različite zadatke koji su povezani s fazom primjene. Neki od uobičajenih zadataka koji se obavljaju na procesorima su izvođenje animacija, simulaciju fizike i mnoge druge, ovisno o vrsti aplikacije. [6]

Sljedeća ključna faza je obrada geometrije, koja se bavi transformacijama, projekcijama i drugim operacijama nad geometrijskim podacima. Ova faza izračunava što treba prikazati, kako to treba biti prikazano i gdje treba biti prikazano. Obrada geometrije obično se izvodi na grafičkoj procesorskoj jedinici (eng. Graphics processing unit, GPU) koja sadrži mnogo programabilnih jezgri i hardverskih komponenti. [6]

Faza rasterizacije obično se događa tako da uzima tri vrha i stvara trokut te zatim određuje sve piksele koji se smatraju unutar tog trokuta. Ovi pikseli se potom prenose u sljedeću fazu. Konačno, faza obrade piksela izvršava program za svaki piksel kako bi odredila njegovu boju i može provesti testiranje dubine kako bi se utvrdilo je li piksel vidljiv. Također može obavljati operacije poput miješanja nove boje s prethodnom bojom. Rasterizacija i faza obrade piksela također se uglavnom provode na grafičkoj procesorskoj jedinici. [6]

Razvojni programer ima potpunu kontrolu nad fazom aplikacije, budući da se obično izvodi na centralnim procesorskim jedinicama. Zbog toga programer ima potpunu slobodu da definiira i prilagodi implementaciju prema potrebama, te čak kasnije može vršiti modifikacije kako bi poboljšao izvedbu. Promjene u ovoj fazi također mogu imati utjecaj na performanse sljedećih faza. [6]

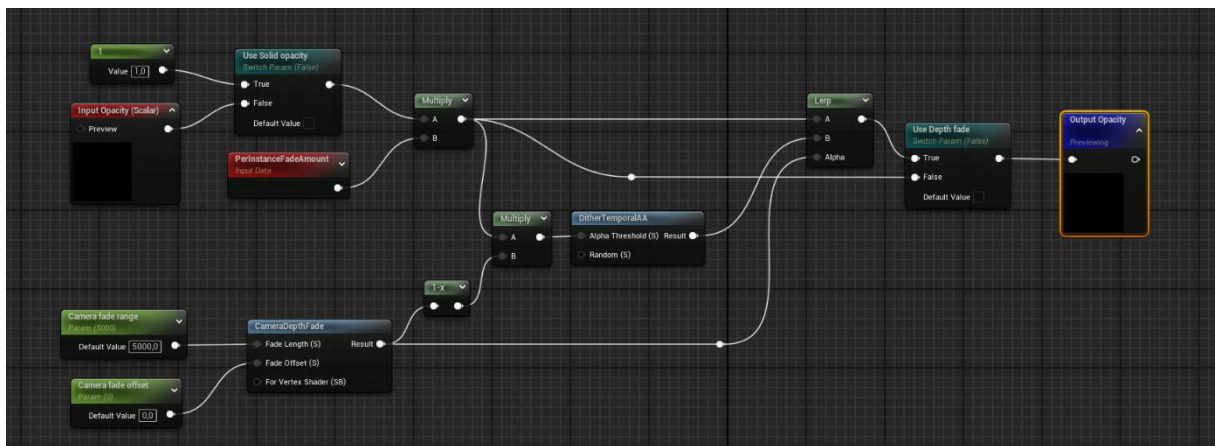
Grafička procesorska jedinica može izvoditi određene aplikacije koristeći poseban način rada poznat kao „compute shader“. U ovom režimu, GPU se tretira kao paralelan procesor, a njegova specifična funkcionalnost za renderiranje grafike se zanemaruje. Nakon završetka faze primjene, geometrija koja se prikazuje prenosi se u fazu obrade geometrije. Ovdje se radi o osnovnim grafičkim entitetima, poput točaka, linija i trokuta, koji bi na kraju mogli biti prikazani na ekranu ili nekom drugom izlaznom uređaju. Ovo je ključni korak u fazi primjene. [6]

Kao rezultat softverske implementacije ove faze, obično se ne dijeli na pod stupnjeve kao što su obrada geometrije, rasterizacija i obrada piksela. Međutim, kako bi se postigla veća učinkovitost, ova se faza često izvodi paralelno na više procesorskih jezgri. U kontekstu dizajna centralnih procesorskih jedinica, ovo se često naziva „superskalarnom arhitekturom“ jer omogućava izvođenje više procesa istovremeno unutar istog vremenskog intervala. [6]

Tok renderiranja predstavlja srce postupka renderiranja u stvarnom vremenu, obuhvaćajući niz koraka koji preuzimaju opis scene i transformiraju ga u vizualno iskustvo.

2.5 BLUEPRINTOVI

Unreal Engine koristi skriptiranje i programiranje kako bi se implementirala logika igre. „Blueprints“ je vizualni programski jezik unutar Unreal Engine-a koji omogućava programerima da stvaraju logiku igre bez pisanja koda. Korisnici mogu koristiti blokove i čvorove da bi definirali ponašanje objekata, interakcije, umjetne inteligencije, animacije i druge elemente igre. [7] Ovaj pristup olakšava iterativno i brzo prototipiranje igara. Osim „blueprintove“, Unreal Engine podržava programski jezik C++, što omogućava naprednije programiranje i prilagodljivost igre.



Slika 6: Primjer Blueprinta

Izvor: Autor

Postoje dvije glavne vrste „blueprinta“: „blueprint“ nivoa i „blueprint“ klase. Svaki nivo igre ima vlastiti „blueprint“ nivoa i nije moguće stvoriti zasebni „blueprint“ nivoa. S druge strane, „Blueprintovi“ klasa se koriste za stvaranje interaktivnih

elemenata igre i mogu se ponovno koristiti na bilo kojem nivou. [8] Na slici 6. prikazan je „blueprint“ klase koji je ispisan u svrhu dizajna tehnike auto materijala.

2.6 VARIABLE

Varijable su sistem koji „Blueprint“ koristi za pohranu podataka. Kao u algebri, gdje koristimo varijablu imena X za pohranjivanje broja, u Unrealu za pohranu podataka koristimo variable. Osim toga, u Unrealu varijable mogu sadržavati i druge vrste podataka osim brojeva. Podaci kao što je tekst. Vrsta varijable određuje koju vrstu podataka može sadržavati.

„Bool“ tip variable se koristi za držanje vrijednosti „True“ ili „False“, to su jedine dvije vrijednosti koje može sadržavati. Zbog toga „Booleov“ tip podataka zauzima vrlo malo prostora u memoriji. [9]

„Byte“, „Integer“, „Integer64“ i „Float“ se koriste za „držanje“ brojeva. Svaki od njih sadrži različite vrste brojeva i kao rezultat toga svaki zauzima različitu količinu prostora u memoriji. U ovom radu koristit će se često naredba „Float“. „Float“ predstavlja broj s pomičnim zarezom. „Float“ se koristi za držanje brojeva koji imaju decimalno mjesto. Može se koristiti za pohranjivanje brojeva koji nisu cijeli, kao što je 2,1 i slično. Zbog toga zahtijeva više memorije nego cijeli broj. [10]

„Name“, „String“ i „Text“ koriste se za pohranu teksta. Najveći tip podataka od ova tri je „Text“. S obzirom na to da on zahtijeva najveću količinu memorije od tri, treba se koristiti samo za svoju specifičnu svrhu, što je pohranjivanje podataka koji će se prikazati na zaslonu igrača. Tip podataka „Text“ koristan je za prikaz teksta jer ima, između ostalog, značajke lokalizacije, što mu omogućuje prikaz teksta na način koji je prilagođen regiji ili jeziku igrača. Podatkovni tip „String“ koristi se za pohranjivanje teksta na kojem se izvršavaju manipulativne funkcije. Ove funkcije uključuju izdvajanje podniza teksta iz većeg dijela, mijenjanje velikih ili malih slova u tekstu, preokretanje teksta i slično. Tip vrste podatka „String“ manji je od „Texta“. Najmanji od ova tri tipa podataka je „Name“. On nema lokalizaciju ili druge značajke koje tip podataka „Text“ ima za prikaz teksta na ekranu, također nema funkcije upravljanja kao tip podataka

„String“. Međutim, zauzima najmanje memorije od sva tri. Zbog toga tip podataka „Name“ bi koristili za bilo koji tekst koji ne zahtijeva značajke tipa podataka „String“ ili „Text“. [10]

„Vector“ je tip podatka koji se koristi za pohranu tri vrijednosti „Float“. Ovo je korisno za definiranje točke u prostoru, RGB vrijednosti ili bilo čega što je definirano s tri vrijednosti. [10]

2.7 SUSTAV KOLIZIJE I FIZIKE

Detekcija kolizije igra ključnu ulogu u svakoj igri jer omogućuje objektima da se međusobno ne prodiru i omogućuje interakciju s virtualnim svijetom na smislen način. Neki naslovi također implementiraju realistične ili polurealistične simulacije dinamike, poznate u industriji igara kao „sustav fizike“, iako bi se termin „dinamika krutog tijela“ često bolje odnosio na taj aspekt, jer se uglavnom bavi kinematikom i dinamikom gibanja krutih tijela, kao i silama i momentima koji upravljaju tim gibanjem. Ovaj sloj je prikazan na slici. [3]

Kolizija i fizika često su međusobno povezani jer se sudari obično rješavaju kao integralni dio procesa integracije fizike i logike za ispunjavanje ograničenja. Danas, rijetko koja razvojna tvrtka za igre razvija vlastiti sustav za koliziju i fiziku. Umjesto toga, obično integriraju softverski razvojni komplet treće strane u svoj „engine“. [3]

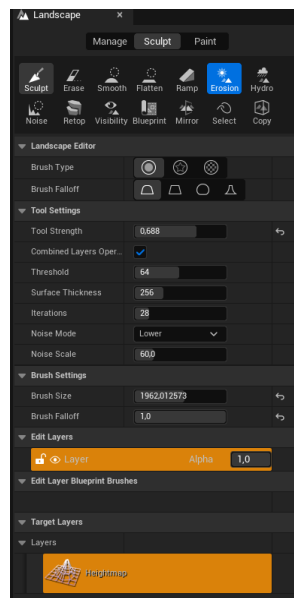
„Havok“ je trenutni standard u industriji igara i odlikuje se bogatim setom značajki, pružajući izvrsne rezultate na svim razinama. Postoji i „engine“ za koliziju i dinamiku „PhysX“, razvijen od strane „NVIDIA-e“, koji je također visoko cijenjen u industriji. Integriran je u Unreal Engine 5 i dostupan je besplatno kao samostalan proizvod za razvoj PC igara. „PhysX“ je prvotno osmišljen kao sučelje za „Ageijin“ fizikalni akcelerator, a danas ga posjeduje i distribuira NVIDIA. Tvrtka je prilagodila „PhysX“ kako bi radio s najnovijim grafičkim procesorima. [3]

3. PRAKTIČNI DIO

Ovaj rad fokusira se na razvoj video igre u Unreal Engine 5 s naglaskom na dizajn grafike kroz upotrebu različitih tehnika za kreaciju sadržaja potrebnog za oživljavanje svijeta igrača. Rezultat je finalno grafičko rješenje sa fokusom na prirodni osjećaj otvorenog svijeta i atmosferu „napuštenosti“ koja prevladava u okolini igrača. Efekt atmosfere se u određenom tonu mijenja ovisno o vremenskom periodu dana koji je prikazan kroz dinamično globalno osvjetljenje. Kako bi postigli ovi efekti koriste se metode navedene u sljedećim poglavljima.

3.1 PEJZAŽNI SUSTAV

Unutar Unreal Engine 5 postoji pejzažni sustav (eng. Landscape) sa nizom alata koji omogućuju izradu različitih tipova nivoa. Na slici 7. prikazan je sustav „Landscape“ koji ima tri radna načina: Upravljanje, Oblikovanje i Bojanje. Svaki od ovih načina pruža različite mogućnosti za interakciju s oblikovanje nivoa.



Slika 7: Pejzažni sustav

Izvor: Autor

Upravljanje (eng. Manage) služi za kreiranja novih pejzaža i prilagođavanje komponenata pejzaža. Također, u režimu upravljanja nalazi se alat za kopiranje pejzaža kako bi kopirali, zalijepili, uvozili i izvozili dijelove pejzaža.

Oblikovanja (eng. Sculpt) služi za modifikaciju pejzaža određenim alatima za uređivanje osnovne visine nivoa (eng. Heightmap). Na slici 8. i 9. prikazani su primjeri modifikacije pejzaža sustavom Oblikovanje. Ovi alati variraju od jednostavnog alata „Sculpt“ koji koristi „kist“ i kontrolu jačine kako bi promijenio snagu kista, do mnogih drugih alata koji primjenjuju složene algoritme kako bi stvorili različite efekte na „heightmapi“, kao što je erozija. Svaki alat posjeduje svoj skup svojstava koji određuju kako će taj alat utjecati na pejzaž. [11]



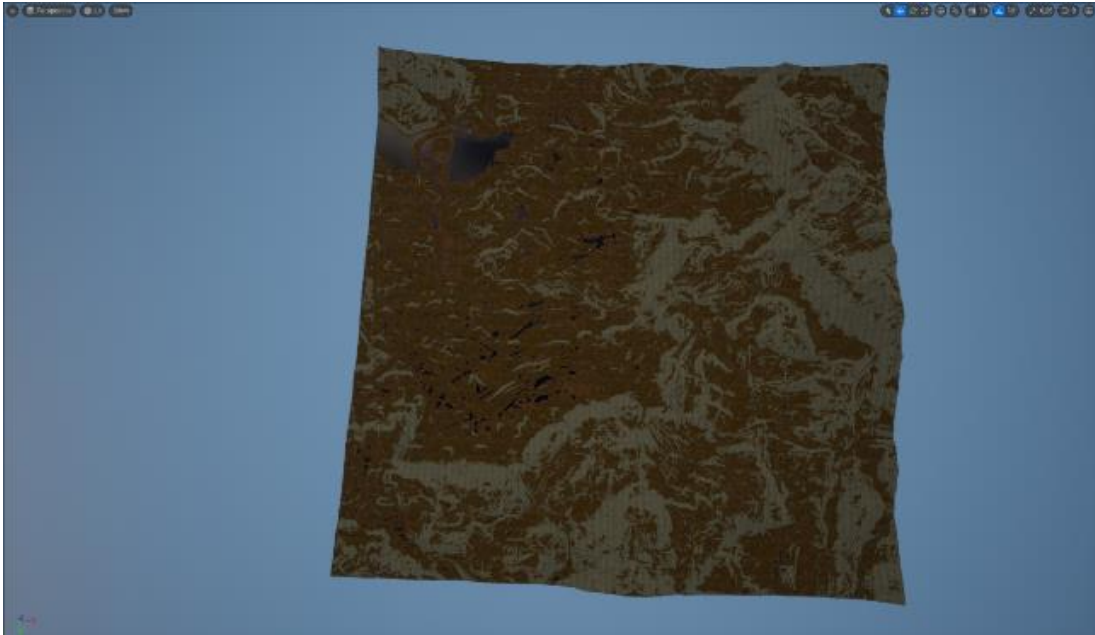
Slika 8 i 9: Primjer modifikacije pejzaža s alatom Oblikovanje

Izvor: Autor

Bojanje (eng. Paint) služi za mijenjanje izgleda dijelova pejzaža slikanjem tekstura direktno na nivou, na temelju slojeva definiranih u materijalu pejzaža (eng. Landscape's Material). Pomicanjem kista po selektiranome nivou ostaje trag zadanog pejzažnog materijala. S ovim alatom iscrtavamo željene teksture na površinu nivoa. Teksture mogu biti materijal koji god mi zadano, na primjer, trava, zemlja, blato, kamen, cesta, voda i slično.

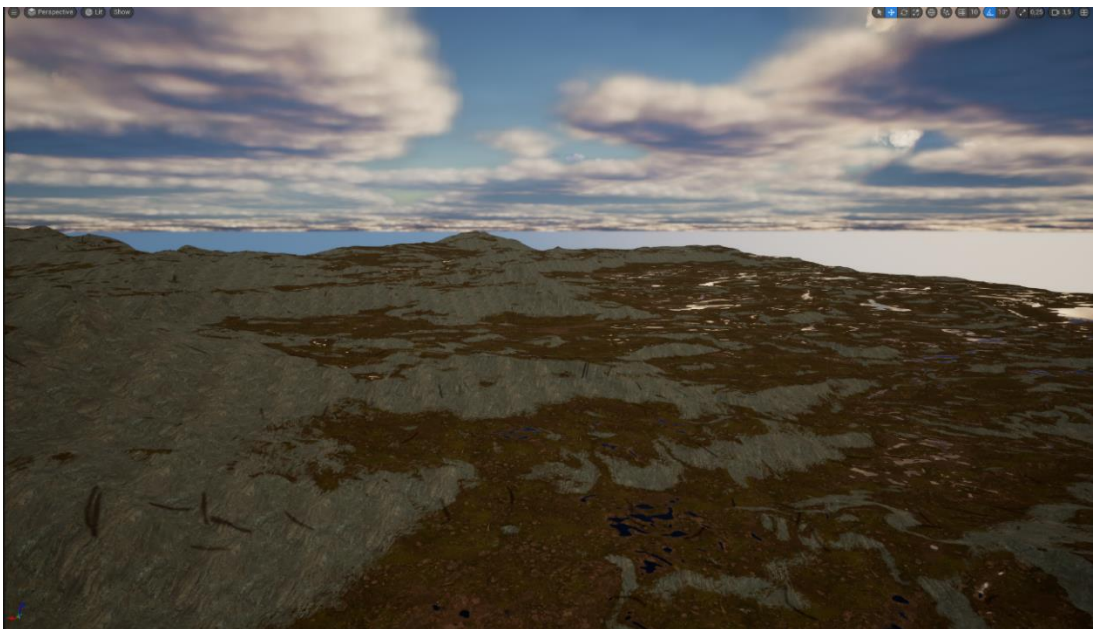
Svaka od ova tri režima upravljanja imaju svoje dodatne postavke koje nam omogućuju složenu izmjenu i modifikaciju pejzaža kroz različite zadane opcije. Zbog toga postoje velike mogućnosti u dizajnu potpuno različitih pejzaža ovisno o potrebama

dizajnera. Na slikama 10. i 11. prikazani su finalni pejzaži nivoa koji su oblikovani sustavom „Landscape“.



Slika 10: Pejzaž cijelog nivoa, prva perspektiva

Izvor: Autor

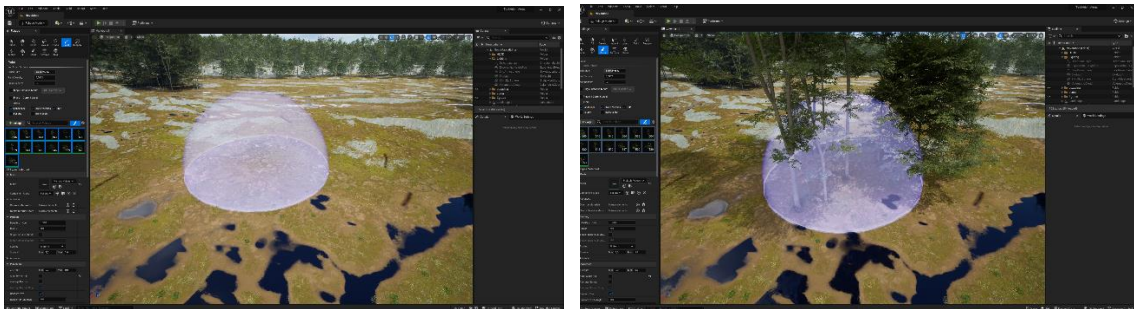


Slika 11: Pejzaž cijelog nivoa, druga perspektiva

Izvor: Autor

3.2 FOLIAGE

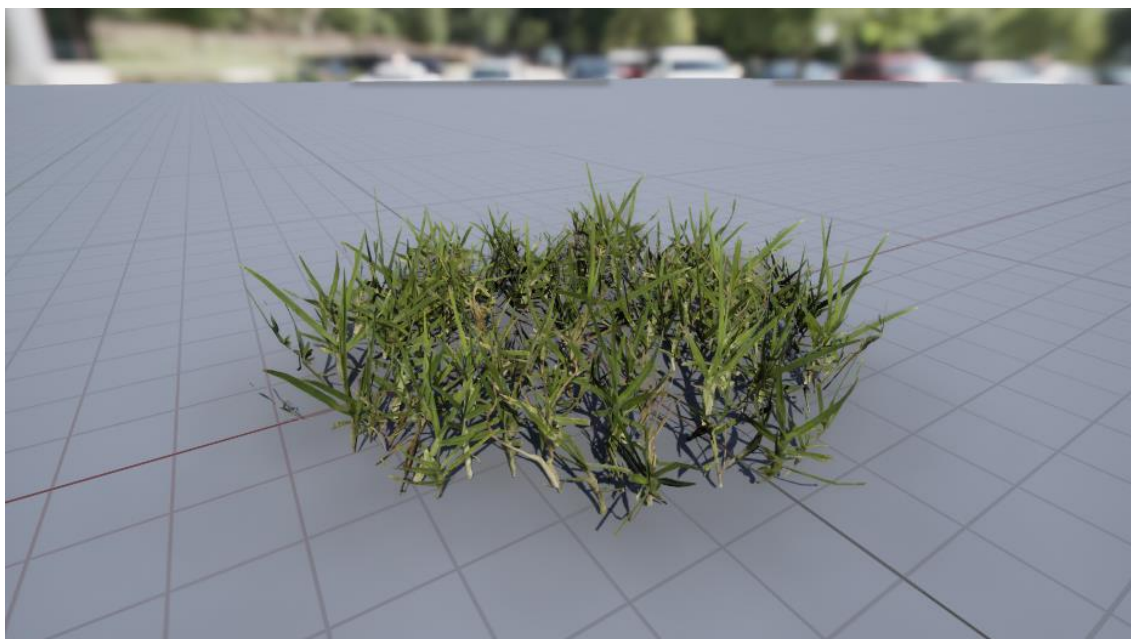
„Foliage“ sustav omogućava jednostavno postavljanje i upravljanje vegetacijom i ostalim objektima u nivou igrice. Postiže se pomoću gotovih modela vegetacije (kao što su drveće, grmlje i trava), ili stvaranjem vlastitih modela. [12] Alati za postavljanje vegetaciju su „Foliage Brush“ i „Foliage Placing“. Na slikama 12. i 13. prikazan je polukrug radnog prostora alata „Foliage Brush“ prije postavljanja i nakon postavljanja vegetacije. Ovisno o postavkama, alat će nasumice unutar polukruga za postavljanje rasporediti vegetaciju kako bi stvorili prirodan izgled. Nakon postavljanja vegetacije možemo prilagoditi postavke kao što su promjena gustoće, smjer rotacije i skaliranja vegetacije. Također, možemo dodati različite vrste vegetacije na istu površinu kako biste postigli raznolikost.



Slika 12 i 13: Uporaba „Foliage“ kista

Izvor: Autor

Na slikama 14. i 15 prikazani su trodimenzionalni „asseti“ trave i biljke koji su korišteni u „foliage“ sistemu za popunjavanje vegetacije mape.



Slika 14: „Foliage“ trave korišten u radu

Izvor: Autor



Slika 15: „Foliage“ biljke maline korišten u radu

Izvor: Autor

3.3 AUTO MATERIJAL

Kroz uporabu „blueprint“ sistema izvedena je složena logika igre koja dopušta razinu dinamike koja u dugoročnom procesu dizajna povelike ubrzava izradu nivoa mape. Korištena tehnika je dinamički auto materijal za izradu pejzaža stvorenih u Unreal Engine-u.



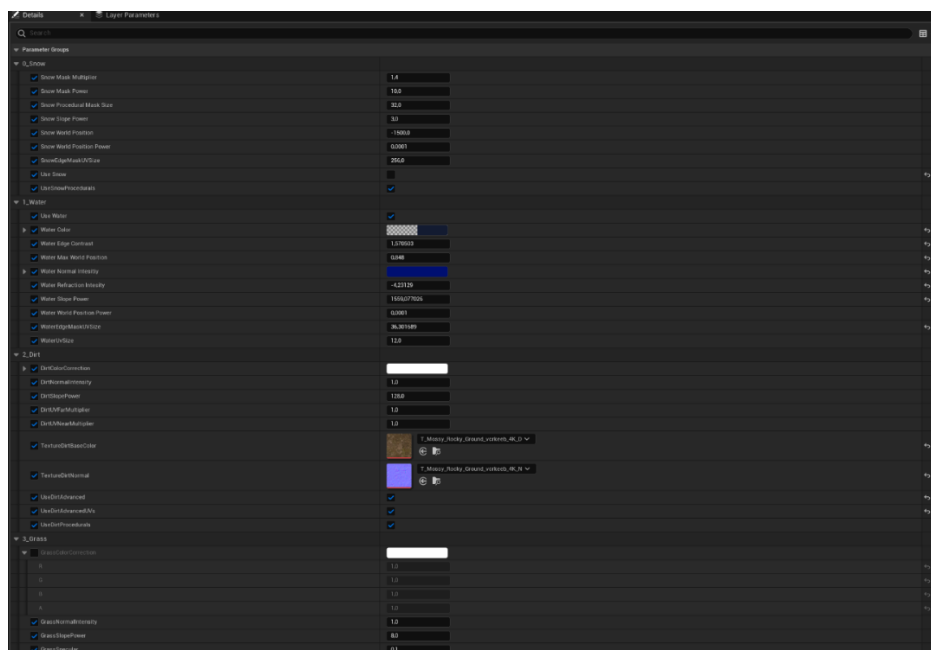
Slika 16 i 17: Promjena parametara dinamičkog auto materijala

Izvor: Autor

Dinamički auto materijal za pejzaž omogućuje promjenu različitih elemenata kao što su teksture, boje i detalji ovisno o faktorima kao što su visina terena i nagib. Za razliku od tradicionalnog pristupa gdje su ti elementi pejzaža statički i potrebna je manualna promjena za svaki postavljeni detalj nivoa, ova metoda dopušta automatsku promjenu tijekom izrade topologije nivoa.

Nakon postavljanja „blueprinta“ klase navedene funkcije kao zadanog materijala nivoa cijela površina poprima zadanu teksturu ovisno o visini skulpturiranih dijelova nivoa. Varijacija i intenzitet svakog elementa odlučena je kroz složenu logiku ispisanih funkcija. Na slici 16.i 17. prikazan je primjer mogućnosti promjene u elementu voda. Ovisno o postavka parametara, voda može biti skroz uklonjena, ili potpuno prevladavati površinom.

Metoda dopušta detaljnu modifikaciju elemenata funkcije. Izvedeni dinamički elementi za izrađeni auto materijala su snijeg, voda, zemlja, trava, kamenje i stijene. Na slici 18. prikazani su elementi funkcije koje je moguće dodatno modificirati kroz grupe parametara koji su određeni za njihovu funkciju. Na primjer, za element voda moguće je promijeniti sljedeće parametre: snaga vodenog nagiba, poziciju vode u relaciji s topografijom, intenzitet vode, boju vode, postojanost vode i UV kanale vode.

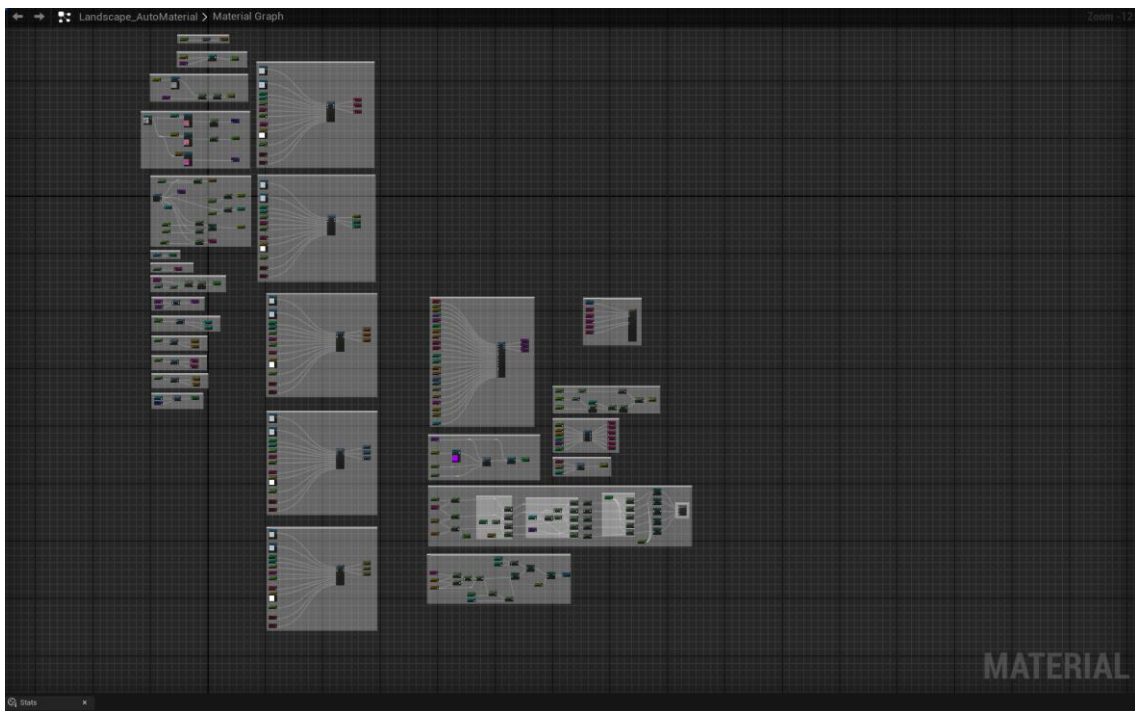


Slika 18: Parametri dinamičkog auto materijala

Izvor: Autor

UV kanal je skup podataka unutar statične mreže koji preslikava svaki od vrhova mreže na koordinate u 2D prostoru. Ova mapiranja definiraju kako se 2D mape teksture omotavaju oko 3D geometrije kada se mreža renderira. UV kanali imaju dvije glavne svrhe u engine-u: definiranje koordinata tekstura u materijalima i pohranjivanje i primjenjivanje svjetlosnih mapa (eng. Lightmaps). Svjetlosna mapa je posebna vrsta teksture koja pohranjuje unaprijed izračunate informacije o osvjetljenju za statične mreže. [13]

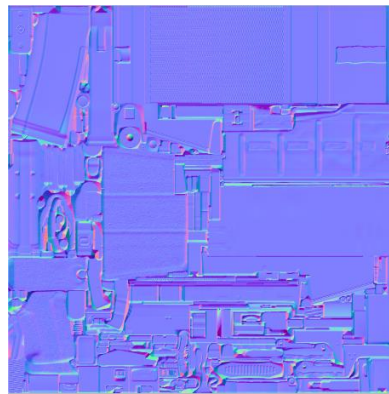
Svaki navedeni parametar elemenata izveden je kroz čvorove (eng. Nodes) koji su objekti kao događaji, pozivi funkcija, operacije kontrole toka, varijable itd. koji se mogu koristiti u grafovima za definiranje funkcionalnosti određenog grafa i „blueprinta“ koji ga sadrži. Svaki tip čvora obavlja jedinstvenu funkciju; međutim, način na koji se čvorovi stvaraju i koriste zajednički je svim čvorovima. To pomaže u stvaranju intuitivnog pristupa pri izradi grafikona čvorova. Na slici 19. prikazan je ispisan sustav „blueprinta“ auto materijala kroz Grafikon čvorova. [14]



Slika 19: „Blueprint“ sistem, Grafikon čvorova

Izvor: Autor

Kako bi „layer“ trava (eng. Grass) dobio svoju funkcionalnost izvedena je „MaterialFunctionCall“ ekspresija koja se koristi za stvaranje ili izvršavanje funkcija materijala koji omogućuje korištenje vanjske funkcije materijala iz drugog materijala ili funkcije. Ulazni i izlazni čvorovi vanjske funkcije postaju ulazi i izlazi čvora poziva funkcije. „MaterialFunctionCall“ je centralni prozor pod nazivom „MF_DefaultLayer“ te predstavlja glavnu i osnovnu zadanu vrijednost layera trava. Izlazni čvorovi „GrassLayer_BaseColor_R“, „GrassLayer_NormalNear_R“ i „GrassLayer_NormalFar_R“ predstavljaju promjenjive parametre za zadane vrijednosti. „Normal“ su posebno stvorene teksture koje sadrže informacije o osvjetljenju modela s kojim se koriste. „Normal“ mapa pomaže stvoriti male detalje koji bi inače bili vrlo hardverski zahtjevni za prikaz u stvarnom vremenu. Na slici 20. prikazana je „Normal“ mapa za korišteni model oružja.



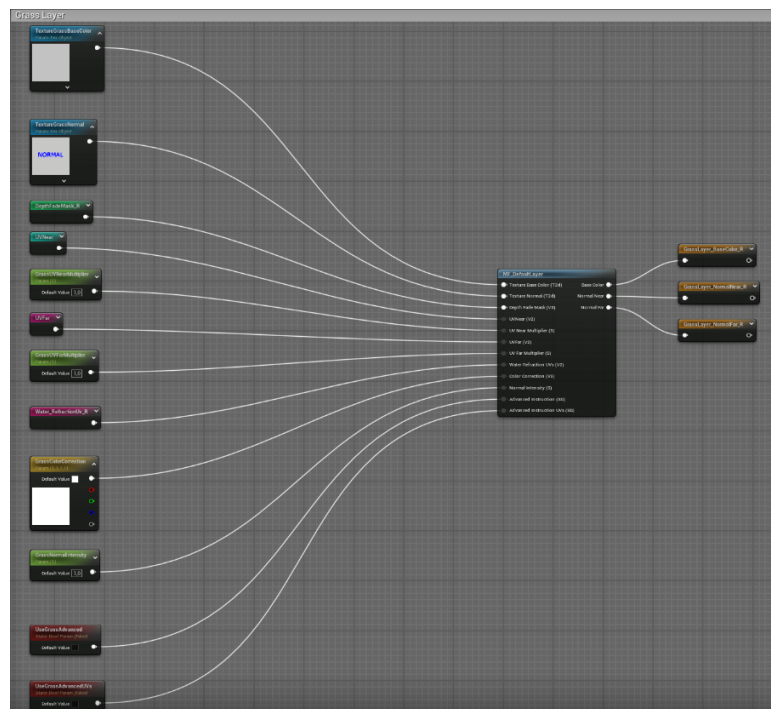
Slika 20: Normal mapa korištenog modela u radu

Izvor: Autor

Korištene ulazne funkcije kako bi trava i ostali elementi dobili funkcionalnost su:

- **Skalarni parametar** (eng. ScalarParameter) daje jednu „float“ (eksponencijalni broj) vrijednost (konstantu) kojoj se može pristupiti i koju možemo promijeniti u materijalnoj instanci (eng. Material Instance) ili pomoću „blueprinta“. [15]

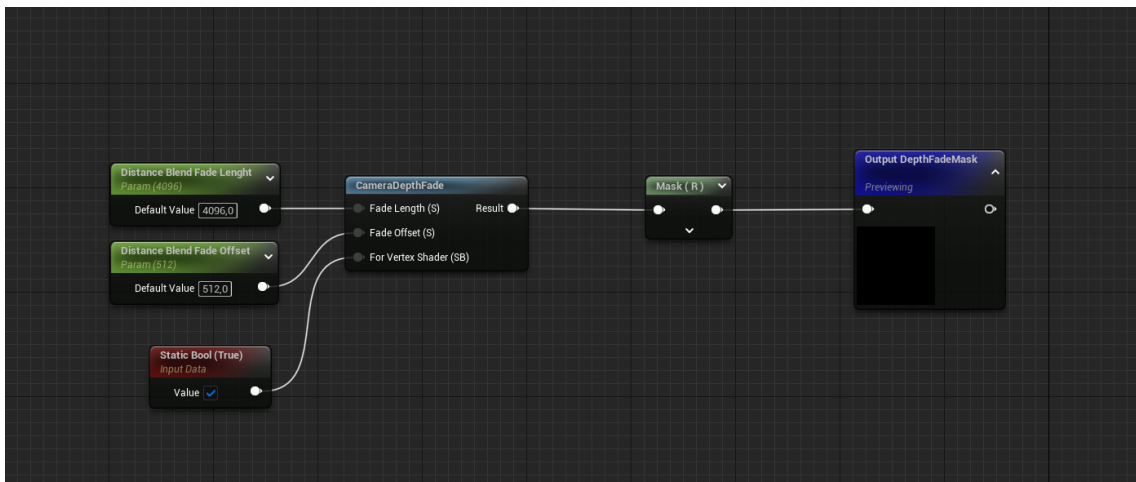
- **Parametar objekta teksture** (eng. TextureObjectParameter) definira parametar teksture i daje objekt teksture. Koristi se za prosljeđivanje parametara teksture u funkciju materijala s unosom teksture. [16]
- **Statički Bool Parametar** (eng. StaticBoolParameter) radi kao „StaticSwitchParameter“ (prima dva ulaza i izlaza. Prvi ako je vrijednost parametra istinita, a drugi ne istinita, te služi kao prekidač), te stvara „bool“ (vrijednost istinita ili lažna) parametar ne implementira prekidač. [17]
- **Vektorski parametar** (eng. VectorParameter) daje četverokanalnu vektorsku vrijednost, četiri konstantna broja. Koristi se za definiranje RGBA boje, gdje je svaki kanal dodijeljen boji (crvena, zelena, plava, alfa). [17]
- **Nazvana preusmjerenja** (eng. Named Reroutes) su „žičani portali“ koji pomažu segmentirati logičke dijelove čvorova. Na slici 21. prikazani su logički dijelovi „UVFar“, „Water_RefractonUV_R“, „UVNear“, „DepthFadeMask_R“. To su dijelovi koji vode na zasebne ekspresije „MaterialFunctionCall“. [14]



Slika 21: Blueprint „layera“ trava (eng. Grass)

Izvor: Autor

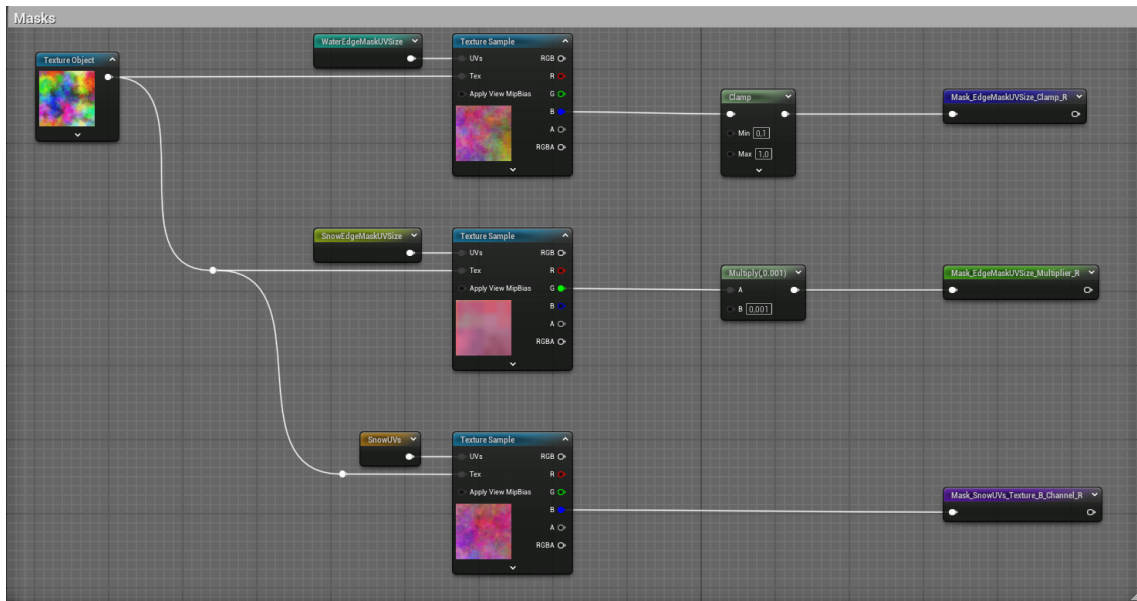
Na slici 22. prikazana je funkcija „DepthFadeMask_R“ koja je izvedena pomoću dva skalarna parametar, statičnog „boola“ i maske. Definirane su joj početne vrijednosti za koliko je jedinica u vrijednosti od 0 do 1 potrebno da „izbljedi“ te koliki je moguć pomak od početne vrijednosti izbljeđivanja. Statični „bool“ za ovaj parametar daje početnu prilagođenu vrijednost „vertex shaderu“ koji mijenja vrijednosti podataka tako da se pojavljuje s drugom bojom, različitim teksturama ili različitim položajem u prostoru (ovisno što mu je zadano). [18]



Slika 22: „Blueprint“ *DepthFadeMask_R*

Izvor: Autor

Kako bi definirali teksturu elementa u auto materijalu koriste se maske. Maska teksture je tekstura u sivim tonovima ili jednom kanalu (R, G, B ili A) teksture koji se koristi za ograničavanje područja efekta unutar materijala. Maska je sadržana unutar jednog kanala, kao što je zeleni i plavi kanal difuzne ili normalne mape. Tehnički, bilo koji kanal bilo koje teksture može se koristiti kao maska teksture.[15] Slika 23. prikazuje korištene funkcije izrađenog „blueprinta“ koji služi kao maske za teksture u auto materijalu.



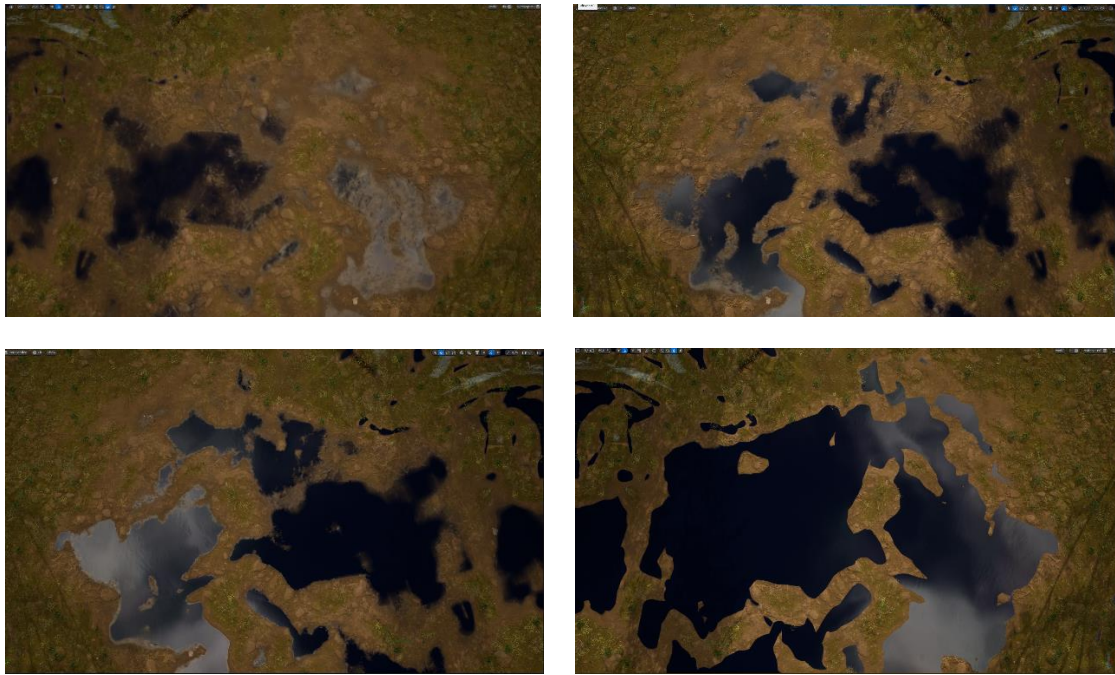
Slika 23: „Blueprint“ Maske za teksture

Izvor: Autor

„Clamp“ funkcija preuzima vrijednosti i ograničava ih na određeni raspon, definiran minimalnom i maksimalnom vrijednošću. Minimalna vrijednost u ovom slučaju je od 0,1 i maksimalna vrijednost od 1,0. To znači da dobivena vrijednosti nikad neće biti manja od 0,1 niti veće od 1,0.

„Multiply“ funkcija uzima dva ulaza (u ovom slučaju jedan je zadan sa 0,001) , množi ih i daje rezultat. Ova funkcija umnožava zadane boje i osnovne boje iz teksture, što rezultira na primjer, tamnijom bojom.

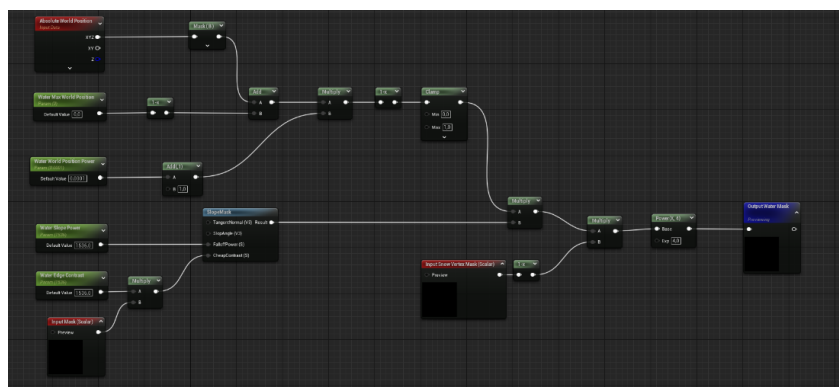
Osim definiranja teksture elemenata maske se koriste i za dinamičnu promjenu materijala. Na slikama 24. , 25. , 26. i 27. prikazano je kako povećanjem, ili smanjenjem vrijednosti dolazi do promjene položaja teksture koja predstavlja određeni element. Ovisno o topografiji nivoa možemo s ovakvim pristupom definirati da li je mapa izrazito stjenovita, potopljena, travnata, snježna i slično.



Slike 24, 25, 26 i 27: Primjer promjene parametara vode u dinamičnom materijalu

Izvor: Autor

Na slici 28. prikazan je „blueprint“ za ispisivanje parametara vode u auto materijalu. Kako bi zadali početnu vrijednost i vrijednost po kojima će se ova funkcija mijenjati koristimo skalarni parametar. Za položaj piksela u prostoru nivoa koristi se funkcija „WorldPosition“. Kako bi povećali, ili smanjili broj prikazanih piksela izvedena je funkcija „Multiply“ u kombinaciji s funkcijom „Power“ s eksponentom četiri. Funkcija „Power“ uzima dva ulaza: osnovnu vrijednost (Base) i eksponent (Exp). Podiže osnovnu vrijednost na potenciju eksponenta i daje rezultat.



Slika 28: „Blueprint“ za ispisivanje parametara vode u dinamičnom materijalu

Izvor: Autor

Ovakva funkcija jedan je od elemenata složene petlje koja „oživljava“ finalno rješenje koje je u ovom slučaju sustav auto materijala. Svaki njezin element daje logičku funkcionalnost koja je na kraju predstavljena kroz pristupačniji način. Unos vrijednosti promjene intenziteta, boje, teksture i drugih sličnih parametara koji su promjenjivi pomakom, ili klikom na „true“ vrijednost bez potrebe za ulaženjem u kod kako bi definirali željene promjene i vrijednosti.

3.4 TEHNIKE OSVIJETLJENA I GEOMETRIJE

Jedna od ključnih značajki Unreal Engine-a je njegova napredna grafika. Podržava različite tehnike osvjetljenja poput globalnog osvjetljenja, dinamičkih sjena, refleksija, volumetrijskog osvjetljenja i efekata čestica. Također podržava visokokvalitetne teksture, visokokvalitetne modele i post-procesiranje koje poboljšava vizualni dojam igre. Dvije moderne tehnike osvjetljenja i prikaza teksture Unreal Engine-a za stvaranje naprednih grafika korištene u ovom radu su Nanite i Lumen.

3.4.1 NANITE

Nanite je tehnologija virtualnog stvaranja i renderiranja koja omogućava prikaz ogromnih količina geometrije s izuzetnom detaljnošću. To znači da možemo stvoriti detaljne vizualne prikaze bez brige o broju poligona ili optimizaciji. [19]



Slika 29: Vizualni sadržaj igre

Izvor: Autor

Nanite koristi metodu virtualne teksture koja omogućava prikazivanje detaljne geometrije u bilo kojem mjerilu. Umjesto tradicionalnih metoda, koje se oslanjaju na različite nivoe detalja (eng. level of detail - LOD) kako bi se prikazivali objekti na daljinu, Nanite prikazuje svaki piksel i poligon u izvornoj rezoluciji, bez obzira na udaljenost. Ova tehnologija postiže izuzetno visoku razlučivost detalja i fotorealističnost, omogućavajući prikazivanje mikro detalja i finih tekstura koje su prije bile ograničene zbog tehničkih ograničenja. Korištenjem ove metode kroz rad prilikom izrade računalne igre postignut je vizualni efekt koji je do sad bio nemoguće izvesti bez velikih gubitaka u performansama izvođenog programa. Na slici 29. prikazan je vizualni sadržaj igre koji koristi Nanite tehnologiju. Zbog nanite-ove mogućnosti kompresije tekstura postignuta je optimizacija u potrošnji memorije koja omogućava efikasno upravljanje velikim količinama podataka što dopušta puno veću razinu detalja vizualnog sadržaja svijeta izvedenog u realnom vremenu. Nanite-ov sistem učitavanja samo geometrije koja je trenutno vidljiva omogućava korištenje detaljne geometrije na cijelom scenariju vizualnog sadržaja rada. Kroz rad korišteni su „asseti“ visoke 8k

(7680 × 4320px) rezolucije koji se pomoću nanite-a skaliraju kroz promjenu gustoće teksture ovisno o udaljenosti promatrača od promatranog objekta. Što je promatrač bliže promatranome objektu razina detalja teksture je bliža maksimalnoj vrijednosti te teksture. Što je promatrač dalje od objekta razina detalja teksture se skalira na manju vrijednost koja ostavlja efekt da je razina detalja nepromijenjena. [19] Na slici 30. prikazan je presjek vizualnog sadržaja igre gdje se vidi gustoća poligona koje nanite koristi za stvaranje detalja.



Slika 30: Prikaz „Nanite“ tehnologije kroz poligone u vizualnom sadržaju igre

Izvor: Autor

Ova tehnologija ima širok spektar primjena u razvoju interaktivne grafike, posebno za igre otvorenog svijeta (eng. open-world games), simulacije i druge projekte koji zahtijevaju visoku razinu detalja. Omogućava dizajnerima i umjetnicima da stvaraju izuzetno detaljne i realistične okoline bez potrebe za ručnom optimizacijom ili smanjenjem kvalitete. [2]

3.4.2 DINAMIČNA GLOBALNA OSVIJETLJENOST

Globalno osvjetljenje (također se naziva samo osvjetljenje) simulira interakcije osvjetljenja s geometrijom i materijalnim površinama kako bi dodalo realistično osvjetljenje scenama. Ova simulacija uzima u obzir apsorpciju i refleksiju materijala s kojima je u interakciji. [20]

Simulacijom načina na koji se svjetlo ponaša u 3D svjetovima upravlja se na jedan od dva načina: korištenjem metoda osvjetljenja u stvarnom vremenu koje podržavaju kretanje svjetla i interakciju dinamičkih svjetala ili korištenjem unaprijed izračunatih (ili pečenih (eng. baked)) informacija o osvjetljenju koje se pohranjuju u teksturama primijenjenim na geometrijske površine. Unreal Engine pruža ova oba načina osvjetljavanja scena i oni nisu isključivi jedan za drugog jer se mogu besprijekorno uklopiti jedan u drugi. [20]

Iako unaprijed izračunato osvjetljenje može rezultirati impresivnim vizualnim efektima, njezina najjača točka istovremeno predstavlja i najveći nedostatak - zahtijeva prethodno izračunavanje. Takvi „offline“ procesi često mogu potrajati dugo vremena. Nije neuobičajeno da za igre treba mnogo sati za izračunavanje svjetlosnih efekata. S obzirom na to da ovi izračuni osvjetljena traju tako dugo, dizajneri su često prisiljeni raditi na više projekata istovremeno kako bi izbjegli gubitak vremena dok čekaju da se izračuni završe. To, s druge strane, može dovesti do pretjeranog opterećenja računalnih resursa i dodatno produljiti vrijeme renderiranja. U nekim situacijama čak nije moguće unaprijed izračunati osvjetljenje, jer se geometrija mijenja tijekom izvođenja ili je generirana dinamički od strane korisnika. Razvijeno je nekoliko metoda za simuliranje globalnog osvjetljenja u dinamičkim okruženjima. Neki od njih ne zahtijevaju prethodno izračunavanje ili fazu pripreme i omogućuju brzu izvedbu svakog „frame-a“.

Metode dinamičkog osvjetljenja u Unreal Engine-u imaju globalna osvjetljenja u stvarnom vremenu za osvjetljavanje scena odbijanjem svjetla iz dinamičkih izvora svjetla. Ovakav pristup omogućuje promjenu rasvjete i automatsko ažuriranje na objekte unutar scene, omogućujući simulaciju prijelaza vremena u danu, ili paljenje i gašenje

svjetla u prostoriji. Budući da se ovi sustavi rasvjete izračunavaju u stvarnom vremenu, potrebno je malo postavljanja da bi funkcionirali.

3.4.3 LUMEN

Lumen je jedna od modernih tehnologija uvedenih u zadnjoj inačici Unreal Engine 5. Ova tehnika se fokusira na globalnu osvjetljenost u igri, posebno na realističnom osvjetljenju i interakciji svjetla s okolinom.



Slika 31: Primjer 1, Promjena osvjetljenja u stvarnome vremenu

Izvor: Autor

Lumen koristi metodu dinamičkog globalnog osvjetljenja, što znači da omogućava svjetlosti da realistično interagiraju s objektima i okolinom u stvarnom vremenu. U svrhu ovoga rada takva metoda ima nekoliko prednosti u procesu dizajniranja. Brži i fleksibilniji radni proces kroz praćenja promjena svjetlosti u sceni i

njihov utjecaj na okolinu u stvarnom vremenu. Ako se objekt pomiče i mijenja položaj ili intenzitet svjetla, Lumen će automatski ažurirati osvjetljenje i sjene u sceni kako bi odražavao te promjene. Prethodne metode osvjetljenja zahtijevale su ručno postavljanje kako bi se postigao željeni efekt što je rezultiralo u puno dužim i otežanim radnim procesom. Druga prednost ove metode je što omogućava detaljno osvjetljenje i atmosfere efekte poput volumetrijskog osvjetljenja i prašine u zraku, pridonoseći stvaranju još uvjerljivijih i realističnijih vizualnih iskustava. [21] Na slikama 31. i 32. prikazane su promjene osvjetljenja nivoa ovisno o kutu izvora svjetlosti.

Dinamično globalno osvjetljenje dopušta atmosfersku manipulaciju koja ovome radu daje dodatnu razinu mogućnosti u realizaciji realističnih razina detalja. Promjena osvjetljenja ovisno o položaju izvora svjetla dopušta dinamičnu reprezentaciju prolaska vremena koja pridonosi efektu realizma. Elementi prikazani u vizualnom sadržaju igre mijenjaju svoja svojstva i dinamiku ovisno o tipu osvjetljenja koje utječe na njih. Oduzimanjem, ili pridodavanjem izvora dinamičnog osvjetljenja ostvaruju se potpuno različiti efekti atmosfere koji prikazuju detaljnost osvjetljenih elemenata na različitim razinama. [2]



Slika 32: Primjer 2, Promjena osvjetljenja u stvarnome vremenu

Izvor: Autor

3.5 ANIMACIJA IGRAČA

U filmu, svaki detalj svake scene pažljivo se planira prije početka snimanja. To uključuje pokrete svakog lika, rekvizita i čak pokrete kamere. Scena se može animirati kao dugotrajan i kontinuiran niz kadrova, a likovi često ne moraju biti animirani kad nisu u kadru kamere.

S druge strane, animacija u video igrama je drugačija. Igra je interaktivno iskustvo, što znači da se unaprijed ne može predvidjeti kako će se likovi kretati i ponašati. Igrač ima potpunu kontrolu nad svojim likom i često ima djelomičnu kontrolu nad kamerom. Čak i likovi koji su vođeni računalom, a nisu igrači, podložni su nepredvidivim radnjama ljudskih igrača. Stoga se animacije u video igrama obično ne stvaraju kao niz uzastopnih slika. Umjesto toga, kretanje likova u igri se razbija na mnogo sitnih pokreta kako bi se omogućila precizna kontrola.

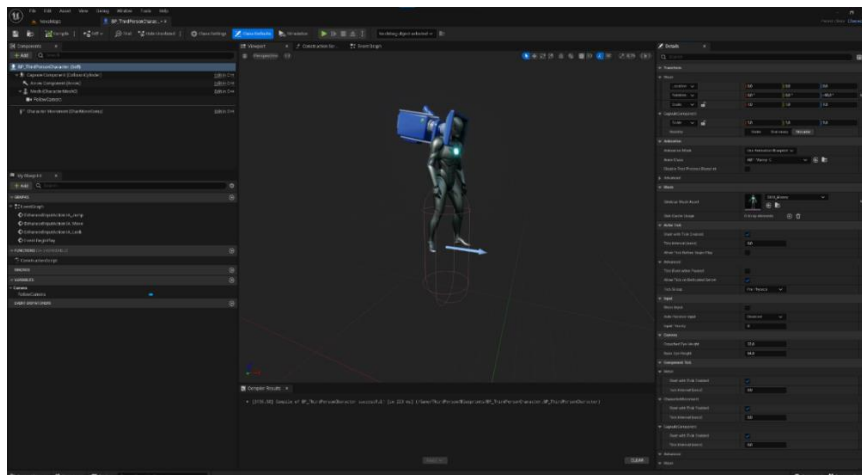
Većina suvremenih 3D igara usredotočena je na likove, često ljudske ili humanoidne, ponekad životinje ili vanzemaljce. Likovi su jedinstveni jer moraju imati fluidne i organske pokrete. Ovo postavlja mnogo novih izazova i tehnoloških zahtjeva, koji idu iznad i izvan onoga što je potrebno za simulaciju i animaciju rigidnih objekata kao što su vozila, projektili, lopte i slično. Postizanje realističnih pokreta likova upravlja se pomoću komponente engine-a poznate kao sustav animacije likova.

Sustav animacije pruža dizajnerima igara skup alata koji se mogu primijeniti na likove i ostale objekte. Gotovo svaki objekt u igri koji nije potpuno nepokretan može iskoristiti prednosti sustava animacije. Vozilo s pokretnim dijelovima, mehanički strojevi s pokretnim dijelovima, pomaci drveća na vjetru, ili zgrada koja eksplodira u igri su sve elementi koji imaju veliku vjerojatnost da su barem djelomično animirani u sustavu animacije.

Unreal Engine koristi „Skeletal Mesh“ kao osnovu za animaciju likova. „Skeletal Mesh“ je prilagodljiva mreža koja omogućuje manipulaciju kako bi se stvorile animacije. Također, „Animation Blueprints“ se mogu primijeniti na „Skeletal Mesh“ kako bi se integrirala logika koja upravlja ponašanjem animacija i njihovim interakcijama unutar nivoa.

„Skeletal assets“ predstavljaju temelj za svaki aspekt animacije sa „Skeletal Meshovima“. „Skeleton editor“ je vizualni alat koji pruža pristup alatima i svojstvima za uređivanje „Skeletal assets“. Ovaj uređivač ima mogućnost manipulacije pojedinačnim dijelovima tj. „kostima“ i „kosturnim strukturama“, te pregledavanje svih animacijskih krivulja i animacijskih obavijesti povezanih s „Skeleton assetom“. [6]

Bez obzira na tehniku koja se koristi za stvaranje animacije, bilo da se temelji na „cel“ animaciji, rigidnoj hijerarhiji ili „skeleton“ animaciji, svaka animacija se odvija tijekom određenog perioda. Da bi se lik doživio kao da se kreće, koristi se iluzija kretanja tako da se tijelo lika postavi u niz diskretnih i statičkih poza, a zatim se te poze prikazuju brzo jedna za drugom, obično s periodom od 30 ili 60 slika u sekundi (eng. Frames per second). U „skeleton“ animaciji, poza kostura izravno kontrolira poziciju vrhova mreže, a postavljanje poza je glavni alat kojim animator daje život svojim likovima. Kostur se postavlja pomoću rotacije, translacije i mogućeg skaliranja zglobova na različite načine. Položaj zglobova definiran je kao položaj, orijentacija zglobova i eventualno skaliranje, u odnosu na neki referentni okvir. Zglobna poza obično se predstavlja putem matrice 4x4, 4x3 ili putem podatkovne strukture SRT (skala, rotacija i translacija u obliku vektora). Poza kostura je skup svih tih poza zglobova i obično se predstavlja kao niz matrica ili SRT-ova. [6]



Slika 33: Položaj kamere za animaciju igrača
Izvor: Autor

Za animaciju kretnje igrača u ovom radu koristi se kamera trećeg lica kako bi se izvela metoda „stvarnog prvog lica“. Ova metoda dopušta animaciju cijeloga tijela u prvom licu, nešto što je inače izgubljeno standardnim pristupom prikaza u prvom licu. Koristi se samo jedan „mesh“ za prikazivanje lika. Na slici 33. prikazan je položaj kamere u odnosu na tijelo igrača. Kamera je pričvršćena na glavu, što znači da ju pokreću animacije tijela. Zbog toga nikada se ne vrše izravne promjene položaja ili rotacije kamere. Cilj je postići realistični oblik animacije gdje igrač u slučaju pomicanja kamere prema dole vidi ostatak svoga tijela, te animacije vidljivih dijelova.

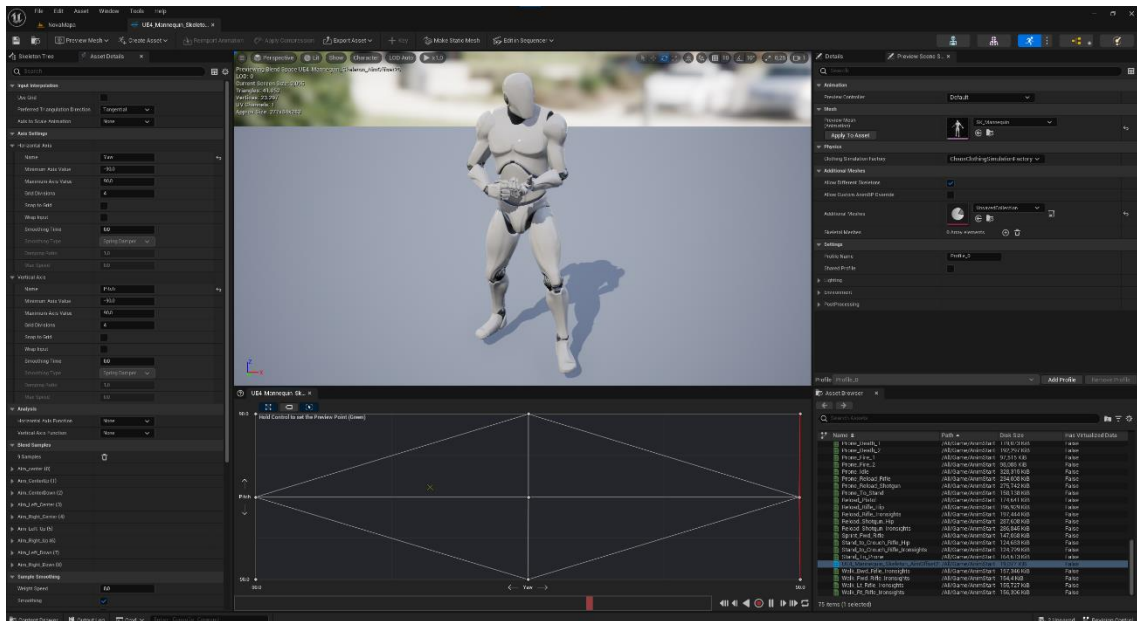
Kako bi napravili animaciju objekta (o ovom slučaju oružja) u ruci igrača koristi se tehnika „Aim Offset“. To je metoda koja pohranjuje niz poza koje se mogu miješati kako bi se liku pomoglo u ciljanju oružja. Kako bi to postigli uzima se sekvenca animacije i dijeli na upotrebljive poze za finalno rješenje, prati se položaj nagiba/pokreta miša igrača kako bi koristili te podatke za određivanje koju kombiniranu pozu primijeniti da se lik pravilno kreće i cilja prema položaju miša. Na slici 34. prikazane su izrađene animacije kao niz poza za „miješane animacije“.



Slika 34: Niz pripremljenih animacija za kombiniranje

Izvor: Autor

Pojam "miješanje animacije" odnosi se na bilo koju tehniku koja omogućuje kombiniranje više od jednog animacijskog isječka kako bi doprinijela konačnom položaju lika. Drugim riječima, miješanje uključuje kombiniranje dviju ili više početnih poza kako bi se stvorila izlazna poza za kostur.



Slika 35: Komponiranje niza animaciju u jednu „miješanu animaciju“

Izvor: Autor

Miješanje obično uključuje kombiniranje dviju ili više poza u istom trenutku i generira izlaz u tom navedenom trenutku. U ovom kontekstu, miješanje se koristi za stvaranje novih animacija kombiniranjem dvije ili više postoje animacije, bez potrebe za ručnim stvaranjem novih animacija. Na primjer, možemo kombinirati animaciju u kojoj lik cilja ulijevo s onom animacijom u kojoj cilja udesno kako bismo postigli željeni kut ciljanja između ta dva ekstrema. Takav pristup animacije se koristi u ovom radu kako bi stvorili tehniku „Aim Offsets“. Na slici 35. prikazan je korišteni sistem komponiranja animacija za izradu miješane animacije. Osim samih animacija tijela ovakav pristup animiranju kroz miješanje se također može koristiti za postizanje glatkih prijelaza između ekstremnih izraza lica, načina kretanja i slično.



Slika 36: Gotov primjer animacije tijela iz perspektive igrača

Izvor: Autor



Slika 37: Gotov primjer animacije tijela u pokretu iz perspektive igrača

Izvor: Autor

Miješanje animacija se također primjenjuje kako bismo pronašli srednju pozu između dviju poznatih poza koje se nalaze u različitim trenucima u vremenu. Ovo je korisno kada želimo odabrati poziciju lika u određenom trenutku koja nije identična ni

jednoj od uzoraka dostupnih u animacijskim podacima. Također, vremensko miješanje animacija može se koristiti za postizanje glatkih prijelaza između dvije animacije, postupno prelazeći iz izvorne animacije u ciljanu tijekom kratkog vremenskog intervala. Takav prijelaz vrlo je koristan za animiranje položaja kao što su čučanj, podizanja oružja, skok i drugih sličnih pomaka kojima je potreban glatki prijelaz. Na slikama 36. , 37. , 38. i 39. prikazane su finalne animacije, tijela u pokretu i držanja oružja iz perspektive igrača.



Slika 38: Primjer 1. animacija nošenja oružja iz perspektive igrača

Izvor: Autor



Slika 39: Primjer 2. animacija nošenja oružja iz perspektive igrača

Izvor: Autor

3.6 FINALNI REZULTAT

Za izradu pejzaža nivoa korištene su različite tehnike koje grade osnovnu bazu vidljivog svijeta igre. Izrađeni auto materijal kroz „Blueprint“ sustav Unreal engine-a početni je element pejzaža na kojemu se naknadno gradi ostatak elemenata kroz „landscape“ i „foliage“ sustave „engine-a“. Promjena postavljenih parametara auto materijala izrazito mijenja dinamiku mape, što ostavlja prostor i mogućnost za buduće promjene, ili prelaze u druge nivoe. Promjena teksture i boje na brz način dozvoljava adaptaciju atmosfere u željenom smjeru dizajnera bez potrebe za izrađivanjem novog nivoa. Promjena godišnjeg doba, ili same tematike nivoa vrlo je jednostavan proces kroz izrađeni auto materijal. Na slici 40. prikazan je gotovi pejzaž nivoa koji je prezentiran kao ljetno godišnje doba. S auto materijalom pejzaž je moguće u potpunosti prekriti samo jednim (ili bilo kojom kombinacijom) od ponuđenih elemenata. To znači da je moguće postaviti nivo koji je u potpunosti prekriven snijegom, stijenom, zemljom, vodom ili travom. Na slici 41. prikazan je primjer nivoa koji je u potpunosti prekriven snijegom.



Slika 40: Primjer 1. Pejzaž nivoa

Izvor: Autor



Slika 41: Primjer 2. Pejzaž nivoa prekriven snijegom

Izvor: Autor

„Foliage“ sistem dozvoljava zamjenu „aseta“ drveća i drugih elemenata bez gubitka zadanih pozicija trenutnih modela. Zamjenom „aseta“ u drugi stil od trenutnog zadanog vrlo je lagan i efikasan proces. Listopadna i mješovita šuma uz zamjenu modela može postati crnogorična šuma, promjena vegetacije bez ponovnog postavljanja položaja modela dozvoljava vrlo brzu alternativu koja postiže promjenu atmosfere, ili tematike nivoa.

Za postavljanje trave i grmova napisan je proceduralni program kroz „blueprint“ sistem, te je povezan s auto materijalom. U slučaju potrebe smanjenja grafičkih elemenata program je moguće potpuno ugasiti, ili odabrati (smanjiti i povećati) stupanj gustoće modela koji se nalaze u njemu. Moguće je odabrati do 25 različitih vrsta modela koji će se proceduralno pojavljivati na pejzažu mape ovisno o elevaciji nivoa. Ovaj tip „aseta“ nalazi se samo na prostorima mape za koji se očekuje da će imati travu, ali kroz „blueprint“ moguće je modificirati područje pojavljivanja. Na slici 42. prikazan je proceduralni prijelaz između vodene i travnate površine.



Slika 42: Primjer 3. Pejzaž nivoa

Izvor: Autor

Kako bi mogli imati modele visoke teksture i rezolucije za izradu vegetacije koristi se tehnologija renderiranja Nanite. Nanite je virtualizirani geometrijski sustav koji koristi novi format mreže i naprednu tehnologiju renderiranja kako bi prikazao detalje na razini piksela i istovremeno veliki broj objekata. Sustav koristi resurse samo na one detalje koji su vidljivi, minimizirajući nepotrebno opterećenje, što nam dopušta korištenje vegetacije u velikoj količini i visokim rezolucijama. Takav sistem dozvoljava puno realističniji izgled grafike u interaktivnoj grafici bez standardnog velikog kompromisa grafičke zahtjevnosti.

Osvjetljenje nivoa izvedeno je pomoću tehnike dinamičkog globalnog osvjetljenja i refleksije „Lumen“. Lumen omogućuje stvaranje difuzne refleksije s beskonačnim odrazima i refleksijama u svim vrstama okruženja, bez obzira na dimenzije. Ova tehnika osvjetljenja je zahtjevnija od tradicionalne tehnike „baking“, ali dopušta dinamičnu promjenu kuta izvora svjetla. Kod „baking“ metode u trenutku kada se osvjetljenje izvrši sjene postaju „zapečene“ i ne mogu se više promijeniti osim ponovnog procesa „bakinga“. Kod dinamičkog osvjetljenja svjetlost i sjene objekta nisu „blokirane“ u jednom položaju, već promjenom pozicije tog objekta sjene i osvjetljenje

automatski će biti ažurirane u novom položaju što olakšava i ubrzava proces dizajna nivoa mape. Ažuriranje u stvarnom vremenu daje dizajnu nivoa efektivan način izrade. Na slici 43. prikazana je drastična promjena osvjetljenja na pejzažu nivoa kroz sustav dinamičnog globalnog osvjetljenja.



Slika 43: Primjer 4. Pejzaž nivoa

Izvor: Autor

Animacija igrača izvedena je tehnikom „miješane animacije“ koja omogućuje glatki prijelaz između različitih tipova pomaka. Kombinacijom niza različitih animacija u jednu miješanu animaciju postiže se tehnika „Aim Offseta“. Ova tehnika vrlo je korisna za višesmjernne pomake koji su izrazito istaknuti u tipu animacije koja je korištena u ovome radu. S obzirom na to da igrač konstantno ima pregled cijelog tijela svog karaktera, te objekt koji se nalazi u ruci igrača je u konstantnom pomaku zajedno sa tijelom karaktera, tijekom pomicanja karaktera u bilo kojem smjeru dolazi do istovremenog pomicanja cijelog tijela i objekta koji se nalazi u ruci igrača. Zbog toga „Aim Offset“ predstavlja idealni pristup animiranju ovakvoga tipa perspektive igrača. „Aim Offset“ je pristup koji je dizajniran kao dodatak već postojećim animacijama.

Kako bi uopće ovaj pristup funkcionirao prvo moramo imati temeljne animacije. U primjeru ciljanja oružja, osnovna animacija obično bi bila neutralna animacija "nišanjenja prema naprijed". Iz ove baze implementiramo puni raspon poza u grafu za animiranje. Broj poza obično ovisi o broju pokreta koji su na potrebni za potpunu animaciju karaktera. Za animaciju pokreta u cijelom prednjem rasponu bez pogleda karaktera „iza sebe“ potrebne su sve točke u kojima je mogući fizički pokret. Na slikama 44. , 45. , 46. i 47. prikazana su četiri pomaka potrebna za definiranje centralne srednje, lijeve i desne, gornje srednje, lijeve i desne, te donje srednje, lijeve i desne točke. Za svaku tu točku potrebna je postojeća animacija položaja karaktera. Zatim iz tih točki u grafu animacija možemo izvesti pokret punog kruga što predstavlja fluidni pomak tijela.



Slika 44, 45, 46, 47: Četiri pomaka za animaciju

Izvor: Autor



Slika 48: Karakter u pokretu

Izvor: Autor



Slika 49: Primjer 5. Pejzaž nivoa

Izvor: Autor



Slika 50: Primjer 6. Pejzaž nivoa

Izvor: Autor

Na slikama 48. , 49. i 50. prikazani su finalni vizualni sadržaji nivoa iz perspektive igrača. Korištene metode za izradu svih elemenata u ovome radu daju rezultat finalnog prikaza cjelovitog rješenja. Kombinirajući elemente sadržaja postiže se željena vizualna atmosfera s fokusom na doživljaj prirodnog osjećaja otvorenog svijeta.

3.7 METODOLOGIJA ISTRAŽIVANJA

Istraživanje je provedeno s ciljem dobivanja generalnog mišljenja korisnika osobnih računala, koji u slobodno vrijeme potencijalno konzumiraju računalne igrice kao zabavni sadržaj . U istraživanju su prikupljeni podaci ispitivanjem tj. provođenjem ankete nad 33 osobe. Anketa je izrađena pomoću alata Google Forms. Ciljevi su bili identificirati elemente grafike koji najviše doprinose doživljaju realizma u igri, te utvrditi kako korisnici doživljavaju i vrednuju realističnu grafiku u igrama. Osim toga kroz niz fotografija, uz zadane parametre, analizirana je kvaliteta realistične grafike napravljene u ovome radu.

Anketa kroz prvih par pitanja provodi profiliranje kako bi utvrdili koja je dominantna dobna skupina, te koji je najčešći tip obrazovanja i razina obrazovanja ispitanika. Kako bi izveli profiliranje postavljena su sljedeća pitanja:

1. Spol? – Muško, Žensko
2. Dob? - 0-18, 19-30, 31-51, 52-70, 70+
3. Razina obrazovanja? - Osnovna škola, Srednja škola, Preddiplomski studij, Diplomski studij, Postdiplomski studij
4. Područje obrazovanja? - Tehničko i informatičko područje, Tjelesno i zdravstveno područje, Matematičko područje, Jezično-komunikacijsko područje, Prirodoslovno područje, Umjetničko područje, Društveno-humanističko područje

Sljedeći dio pitanja utvrđuje osnovne podatke kao što su: tjedni period korištenja računala i igrica, poznavanje korištenog programa u radu i da li prepoznaju koji game engine je korišten za izradu igrica koje konzumiraju. Cilj je bio bolje razumjeti o kakvoj se grupi ispitanika radi, te da li ta grupa prepoznaje korištene alate za izradu medija koji konzumiraju. Postavljena pitanja su sljedeća:

1. Koliko često koristite računalo? – nikada, jednom tjedno, par puta tjedno, svaki dan
2. Da li igrate igrice? - da, ne
3. Koliko često igrate igrice? - nikada, jednom tjedno, par puta tjedno, svaki dan
4. Da li vam je poznat pojam? - "Game engine" nikada nisam čuo/la, čuo/la sam ali nisam siguran/na što je, čuo/la i znam što je otprilike, da i znam što je
5. Da li primjećujete, ili se pitate koji je Game engine korišten u igrici koju u tom trenutku igrate? - Nikada, rijetko, ponekad, često, uvijek

Kako bi identificirali elemente grafike koji najviše doprinose doživljaju realizma u igri, te utvrditi kako korisnici doživljavaju i vrednuju realističnu grafiku u igrama postavljena su sljedeća pitanja:

1. Da li vam je bitna grafika u izboru igrice? - Da, Da, ali nije prioritet, ne

2. Da li vam je bitna grafika u odnosu na cijenu igrice? - Da, Da, ali nije prioritet, ne
3. Volite li Indie igre (igre neovisnih programera, bez velikih izdavača iza sebe)? - Da, Ne, Nije mi bitno tko je izdavač
4. Da li biste bili spremni plaćati više za igru koja ima izvanrednu grafiku u usporedbi s igrama koje su vizualno manje impresivne? - Da, Vjerojatno da, Nisam siguran/na, Vjerojatno ne, Ne
5. Da li smatrate da grafika u igrama može unaprijediti ukupno iskustvo igranja? - Da, Vjerojatno da, Nisam siguran/na, Vjerojatno ne, Ne
6. Da li preferirate realističnu ili stiliziranu grafiku u igrama? - Realističnu, Stiliziranu, Nisam siguran/na
7. Što vam najviše privlači pažnju u pogledu grafike u igri? – Animacije, Kvaliteta teksture, Kvaliteta osvjetljena, Sve navedeno

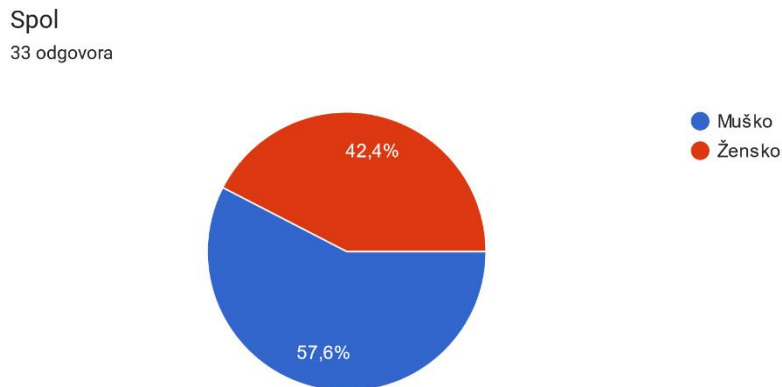
Za finalni dio ankete prikazane su tri fotografije iz rada. Ispitanici su trebali proučiti fotografije, zatim odgovoriti na zadana pitanja o grafičkoj kvaliteti predstavljenih primjera iz igre. Odgovori su bili izvedeni davanjem ocjene za svaki ponuđeni parametar grafičkog svojstva igre. Ponuđene ocjene prikazanih primjera su bile u rasponu od – „vrlo mala, mala, srednja, velika, vrlo velika“. Postavljeni uvjet iznad svake zadane slike je bio sljedeći:

1. Odgovorite na sljedeća pitanja o slici 1. davanjem ocjene od vrlo mala, do vrlo velika za svaki ponuđeni opis. - Grafička privlačnost, Realističnost scene, Kvaliteta tekstura, Kvaliteta osvjetljena, Osjećaj umerzivosti
Ponuđene ocjene su – Vrlo mala, mala, srednja, velika, vrlo velika

Budući da se radi o tehnologiji i tržištu koje je postalo izrazito popularno među širom publikom tek u posljednjih deset godina, bilo je nužno prvobitno istražiti koja dobna skupina pretežito sudjeluje u ovom istraživanju. Nadalje, ključni podatak koji treba uzeti u obzir je obrazovna pozadina jer se radi o specifičnoj tehnologiji koja je više naglašena u određenim područjima kao što su tehničke, tehnološke i umjetničke discipline. To sugerira da osobe s obrazovanjem iz tih područja vjerojatno imaju dublje razumijevanje tematike koja se istražuje u anketi.

3.4. ANALIZA I REZULTATI ANKETE

Ciljem dobivanja sveobuhvatnog uvida u istraživanu temu, svakom ispitaniku je na početku ankete postavljen set profilirajućih pitanja, počevši od spola. U anketi su sudjelovali ispitanici oba spola skoro u ravnom omjeru, s 57,6% muških i 42,4% ženskih ispitanika (Slika 51).

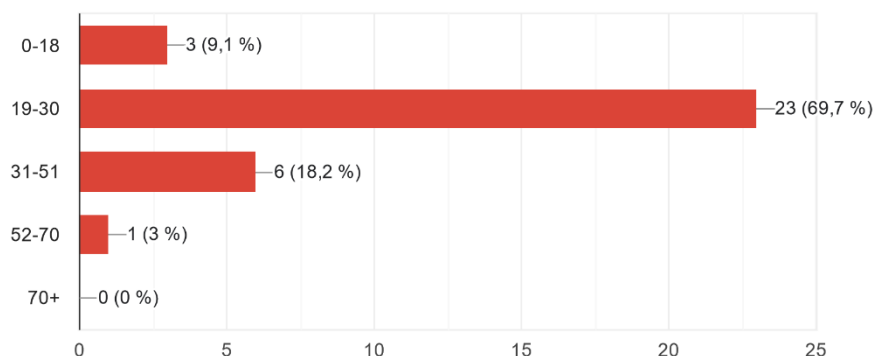


Slika 51: Anketa - Spol

Izvor: Autor

Sljedeće pitanje se odnosi na dob ispitanika. Rezultati istraživanja pokazuju da većina ispitanika (69,7%) spada u dobnu skupinu od 19 do 30 godina. Nakon toga, druga najbrojnija dobna skupina obuhvaća ispitanike u dobi od 31 do 51 godinu (18,2%). Najmanje zastupljene dobne skupine su one u dobi od 0 do 18 godina (9,1%) i od 52 do 70 godina (3%)(Slika 52).

Dob
33 odgovora

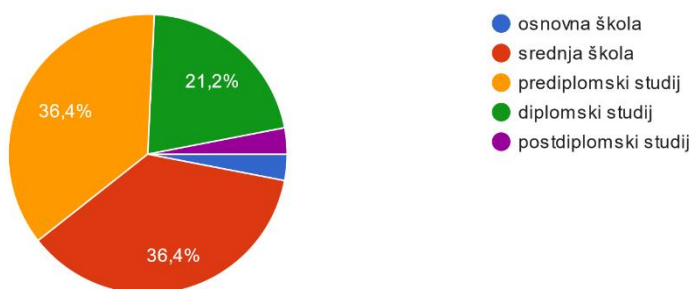


Slika 52: Anketa - Dob

Izvor: Autor

Što se tiče obrazovne razine ispitanika, rezultati pokazuju da 3% ispitanika ima osnovnoškolsko obrazovanje, 36,4% ima srednjoškolsko obrazovanje, isto toliko (36,4%) je završilo preddiplomski studij, 21,2% ispitanika ima diplomski stupanj, dok 3% ispitanika ima postdiplomski stupanj (Slika 53).

Razina obrazovanja
33 odgovora



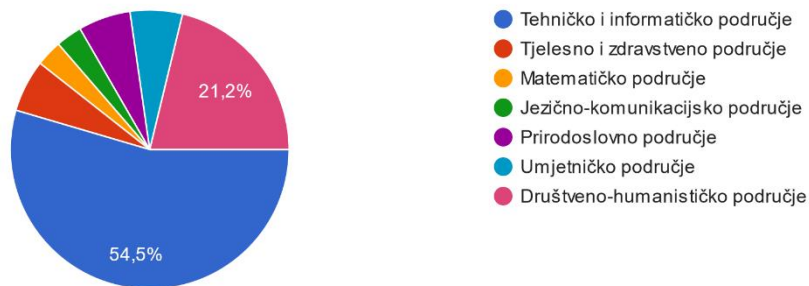
Slika 53: Anketa – Razina obrazovanja

Izvor: Autor

U narednom pitanju ispitanici su navodili svoje obrazovno područje, a rezultati su pokazali da se Tehničko i informatičko područje izdvaja kao najdominantnije s

udjelom od 54,5%. Nakon toga, Društveno-humanističko područje čini 21,2% ispitanika, dok slijede Umjetničko područje, Prirodoslovno područje i Tjelesno i zdravstveno područje s udjelom od 6,1%. Matematičko područje i jezično područje imaju po 3% udjela (Slika 54).

Vaše područje obrazovanja
33 odgovora



Slika 54: Anketa – Područje obrazovanja

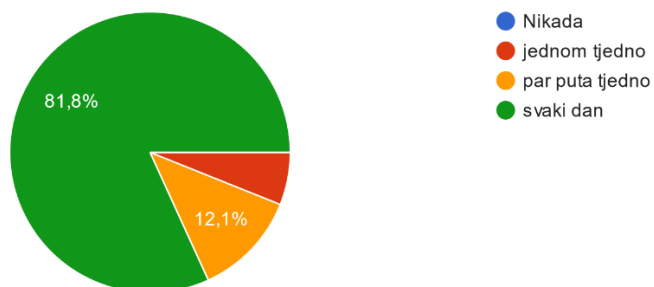
Izvor: Autor

Kako bi dobili uvid u razinu razumijevanja zadane teme u anketi, sljedeća postavljena pitanja usmjerena su na definiranje koliko su ispitanici upoznati s pojmom „game engine“ i koriste li računala i video igre kao svoj izbor zabave.

Početno osnovno pitanje je bilo o učestalosti korištenja računala. Velika većina ispitanika, točnije 81,8%, izjavila je da koristi računalo svakodnevno, dok je 12,1% reklo da ga koristi nekoliko puta tjedno. Jednom tjedno računalo koristi 6,1% ispitanika, dok nijedan ispitanik (0%) nije naveo da nikada ne koristi računalo (Slika 55).

Koliko često koristite računalo

33 odgovora



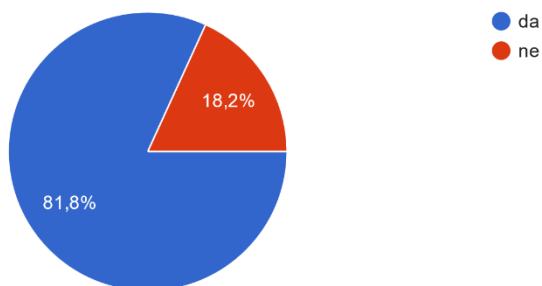
Slika 55: Anketa – Učestalost korištenja računala

Izvor: Autor

Sljedeće pitanje se odnosilo na to igraju li ispitanici video igre. Odgovor je bio da 81,8% ispitanika igra video igre, dok 18,2% ne igra (Slika 56).

Da li igrate igrice

33 odgovora



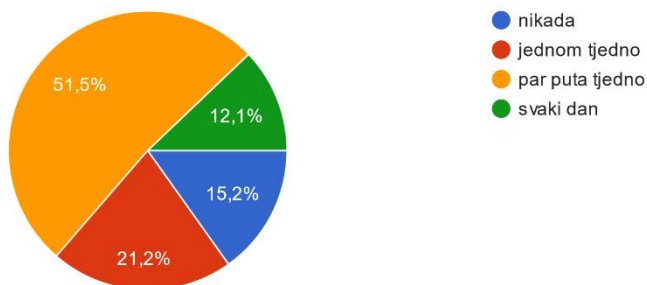
Slika 56: Anketa – Konzumacija igrice

Izvor: Autor

Nadalje, ispitanima je postavljeno pitanje o učestalosti njihovog igranja video igara, a rezultati su pokazali da čak 84,8% korisnika igra video igre svaki tjedan. Od toga, 51,5% igra video igre nekoliko puta tjedno, 21,2% igra jednom tjedno, 15,2% nikada ne igra video igre, dok 12,1% igra svakodnevno (Slika 57).

Koliko često igrate igrice

33 odgovora



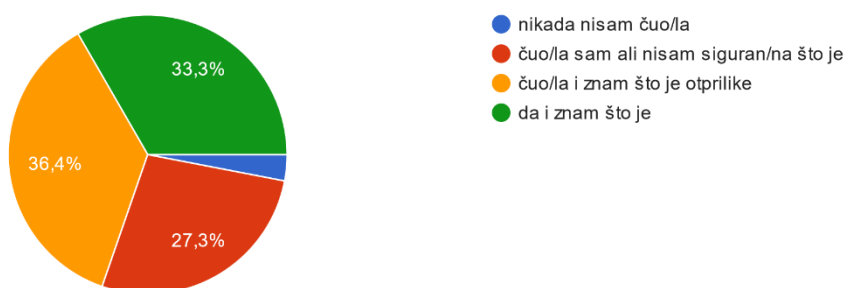
Slika 57: Anketa – Učestalost igranja igrice

Izvor: Autor

Za utvrđivanje razine poznavanja pojma "game engine" među ispitanicima postavljena su dva pitanja. Prvo pitanje je bilo da li su upoznati s pojmom "game engine". Rezultati su pokazali da je 33,3% ispitanika odgovorilo da su upoznati s pojmom i znaju što on znači, 36,4% je reklo da su čuli za taj pojam i znaju otprilike što označava, 27,3% je čulo za taj pojam, ali nisu sigurni u njegovo značenje, dok je 3% odgovorilo da nikada nisu čuli za taj pojam (Slika 58).

Da li vam je poznat pojam "Game engine"

33 odgovora

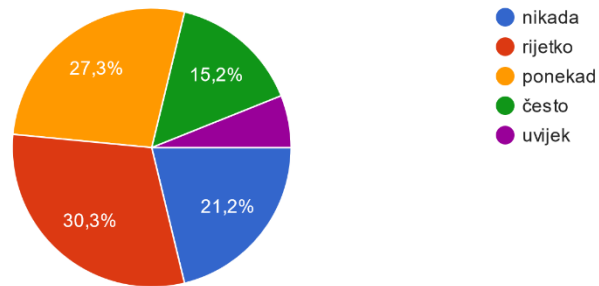


Slika 58: Anketa – Poznavanje pojma „Game engine“ pitanje 1.

Izvor: Autor

Drugo postavljeno pitanje s ciljem utvrđivanja razine poznavanja pojma "game engine" bilo je: "Primjećujete li ili se pitate koji je game engine korišten u igrici koju igrate?" Odgovori ispitanika bili su sljedeći: nikada - 21,2%, rijetko - 30,3%, ponekad - 27,3%, često - 15,2%, i uvijek - 6,1% (Slika 59).

Da li primjećujete, ili se pitate koji je Game engine korišten u igrici koju trenutno igrate?
33 odgovora



Slika 59: Anketa – Poznavanje pojma „Game engine“ pitanje 2.

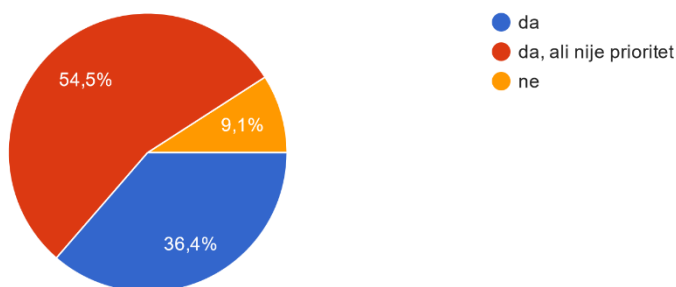
Izvor: Autor

Niz sljedećih pitanja bio je usmjeren na grafička obilježja video igre s ciljem utvrđivanja važnosti grafičkih svojstava igre i njihovog utjecaja na potencijalnu kupnju ispitanika.

Prvo pitanje u nizu se odnosilo na važnost grafike pri odabiru video igre. Rezultati istraživanja su pokazali da je 36,4% ispitanika izjavilo da im je grafika bitna pri izboru igre, 54,5% je reklo da im je važna, ali nije presudna, dok je 9,1% navelo da im grafika nije bitna (Slika 60).

Da li vam je bitna grafika u izboru igrice

33 odgovora



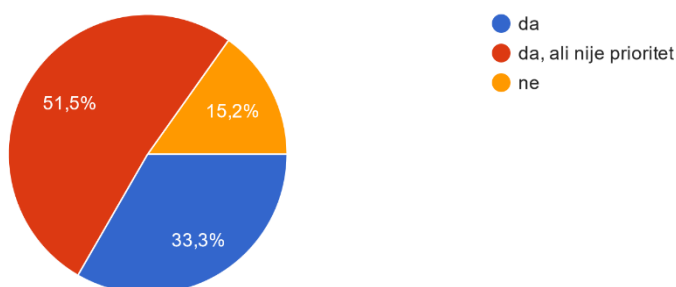
Slika 60: Anketa – Važnost grafike u izboru igre

Izvor: Autor

Drugim pitanjem se istraživala važnost grafike u odnosu na cijenu video igre. Rezultati su pokazali da je 51,5% ispitanika izjavilo da im je grafika važna, ali nije prioritet, 33,3% je reklo da im je važna grafika u odnosu na cijenu, dok je 15,2% odgovorilo da im grafika nije važna u tom kontekstu (Slika 61).

Da li vam je bitna grafika u odnosu na cijenu igrice

33 odgovora



Slika 61: Anketa – Važnost grafike u odnosu na cijenu igrice

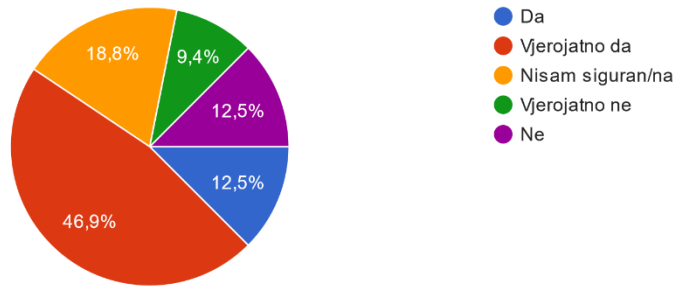
Izvor: Autor

Trećim postavljenim pitanjem o grafičkoj važnosti istraživalo se da li bi ispitanici bili voljni platiti više za igru koja nudi izvanrednu grafiku u usporedbi s igrama koje su vizualno manje impresivne. Rezultati su pokazali da je 12,5% ispitanika

odgovorilo s "da", 46,9% s "vjerojatno da", 18,8% nije bilo sigurno, 9,4% je reklo "vjerojatno ne", dok je 12,5% odgovorilo s "ne" (Slika 62).

Da li biste bili spremni plaćati više za igru koja ima izvanrednu grafiku u usporedbi s igrama koje su vizualno manje impresivne?

32 odgovora



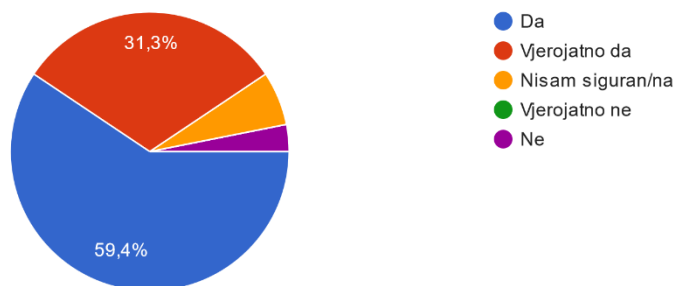
Slika 62: Anketa – Važnost grafike u odnosu na cijenu igrice

Izvor: Autor

Četvrtim pitanjem u nizu istraživalo se mišljenje ispitanika o tome može li grafika u igrama poboljšati ukupno iskustvo igranja. Rezultati su pokazali da je 59,4% ispitanika odgovorilo s "da", 31,3% s "vjerojatno da", 6,3% nije bilo sigurno, dok je 3,1% odgovorilo s "ne" (Slika 63).

Da li smatrate da grafika u igrama može unaprijediti ukupno iskustvo igranja?

32 odgovora



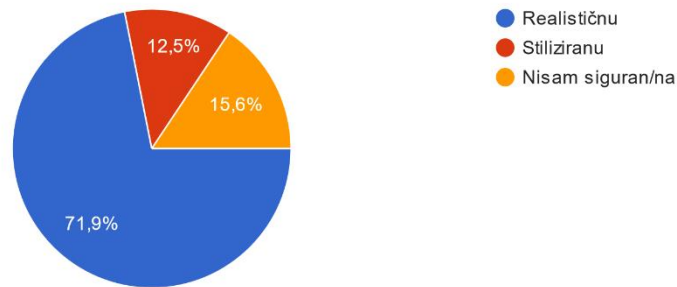
Slika 63: Anketa – Važnost grafike u odnosu na cijenu igrice

Izvor: Autor

Peto pitanje se odnosilo na preferencije u vezi s grafikom u igrama i pitalo je da li ispitanici više preferiraju realističnu ili stiliziranu grafiku. Rezultati su pokazali da je 71,9% ispitanika izrazilo preferenciju prema realističnoj grafici, 15,6% nije bilo sigurno, dok je 12,5% preferiralo stiliziranu grafiku (Slika 64).

Da li preferirate realističnu ili stiliziranu grafiku u igrama?

32 odgovora



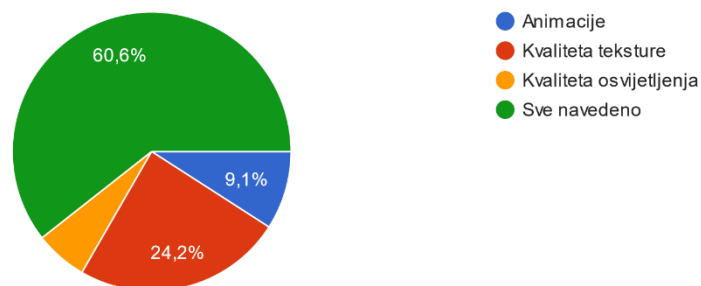
Slika 64: Anketa – Preferiranje tipa grafike

Izvor: Autor

Šesto i posljednje pitanje u ovom nizu o važnosti grafike istraživalo je što ispitanicima najviše privlači pažnju u vezi grafike u igri. Rezultati su pokazali da je 9,1% ispitanika navelo animacije, 24,2% kvalitetu tekstura, 6,1% kvalitetu osvjetljenja, dok je čak 60,6% ispitanika odabralo "sve navedeno" (Slika 65).

Šta vam najviše privlači pažnju u pogledu grafike u igri?

33 odgovora

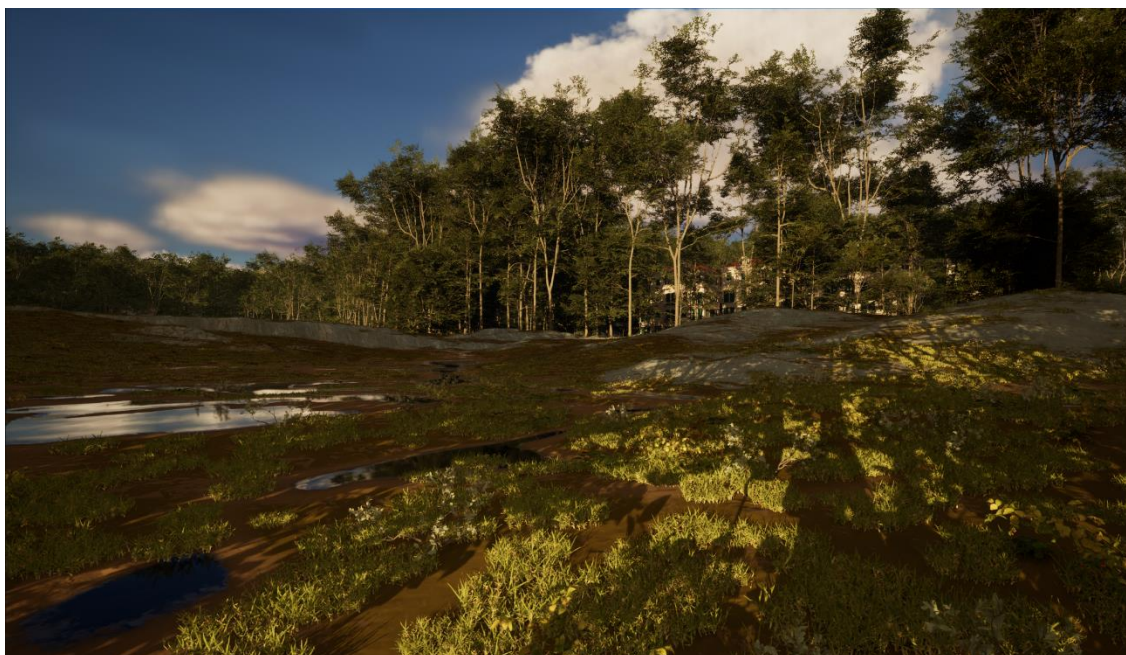


Slika 65: Anketa – Što privlači najveću pažnju u grafici

Izvor: Autor

U završnom dijelu ankete prikazane su tri fotografije iz igre. Ispitanici su bili pozvani da pažljivo prouče te fotografije i nakon toga da odgovore na postavljena pitanja o grafičkoj kvaliteti prikazanih primjera iz igre. Odgovori su se sastojali od ocjena za različite parametre grafičkih svojstava igre, uključujući „Grafičku privlačnost,“ „Realističnost scene,“ „Kvalitetu tekstura,“ „Kvalitetu osvjetljenja,“ i „Osjećaj imerzivnosti.“ Ispitanicima je ponuđena ljestvica za ocjenjivanje tih parametara, koja se sastojala od sljedećih opcija: vrlo niska kvaliteta, niska kvaliteta, srednja kvaliteta, visoka kvaliteta i vrlo visoka kvaliteta.

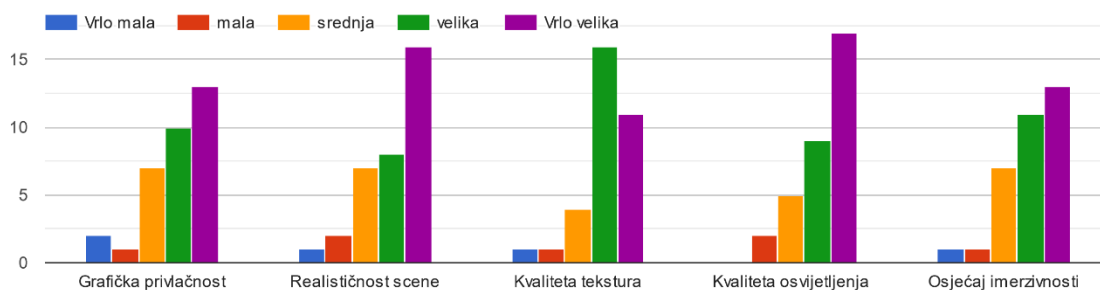
Prvu sliku (Slika 66), 13 ispitanika je ocijenilo „Grafičku privlačnost“ kao vrlo visoku, 10 ispitanika kao visoku, 7 ispitanika kao srednju, 1 ispitanik kao nisku i 2 ispitanika kao vrlo nisku. Što se tiče „Realističnosti scene“, 16 ispitanika je ocijenilo kvalitetu kao vrlo visoku, 8 ispitanika kao visoku, 7 ispitanika kao srednju, 2 ispitanika kao nisku i 1 ispitanik kao vrlo nisku. Za „Kvalitetu tekstura“, 11 ispitanika je ocijenilo kvalitetu kao vrlo visoku, 16 ispitanika kao visoku, 4 ispitanika kao srednju, 1 ispitanik kao nisku i 1 ispitanik kao vrlo nisku. Što se tiče „Kvalitete osvjetljenja“, 17 ispitanika je ocijenilo kvalitetu kao vrlo visoku, 9 ispitanika kao visoku, 5 ispitanika kao srednju i 2 ispitanika kao nisku. Konačno, za „Osjećaja imerzivnosti“, 13 ispitanika je ocijenilo kvalitetu kao vrlo visoku, 11 ispitanika kao visoku, 7 ispitanika kao srednju, 1 ispitanik kao nisku i 1 ispitanik kao vrlo nisku (Slika 67).



Slika 66: Anketa – slika 1 iz ankete

Izvor: Autor

Odgovorite na sljedeća pitanja o slici 1. davanjem ocjene od vrlo mala, do vrlo velika za svaki ponuđeni opis



Slika 67: Anketa – ocjene slike 1 iz rada

Izvor: Autor

Za drugu sliku (Slika 68), 17 ispitanika je ocijenilo „Grafičku privlačnost“ kao vrlo visoku, 9 kao visoku, 6 kao srednju i 1 kao nisku. Što se tiče „Realističnosti scene“, 18 ispitanika je ocijenilo kvalitetu kao vrlo visoku, 12 kao visoku, 2 kao srednju i 1 kao nisku. Za „Kvalitetu tekstura“, 14 ispitanika je ocijenilo kvalitetu kao vrlo visoku, 14

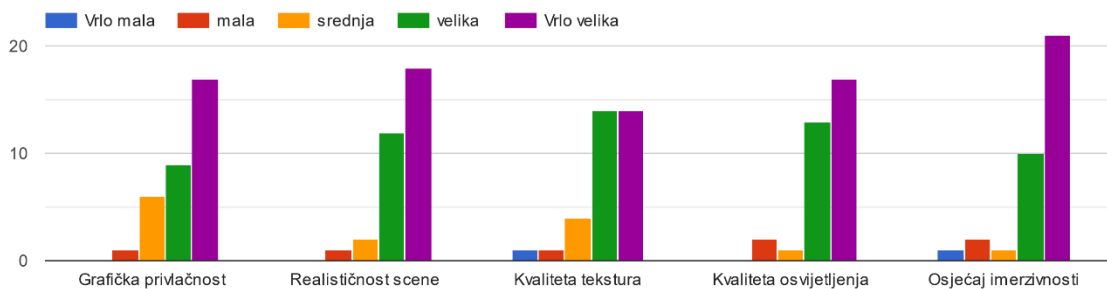
kao visoku, 4 kao srednju, 1 kao nisku i 1 kao vrlo nisku. Za „Kvalitete osvjetljenja“, 17 ispitanika je ocijenilo kvalitetu kao vrlo visoku, 3 kao visoku, 1 kao srednju i 2 kao nisku. Za „Osjećaj imerzivnosti“, 21 ispitanik je ocijenilo kvalitetu kao vrlo visoku, 10 kao visoku, 1 kao srednju, 2 kao nisku i 1 kao vrlo nisku (Slika 69).



Slika 68: Anketa – slika 2 iz ankete

Izvor: Autor

Odgovorite na sljedeća pitanja o slici 2. davanjem ocjene od vrlo mala, do vrlo velika za svaki ponuđeni opis



Slika 69: Anketa – ocjene slike 2 iz rada

Izvor: Autor

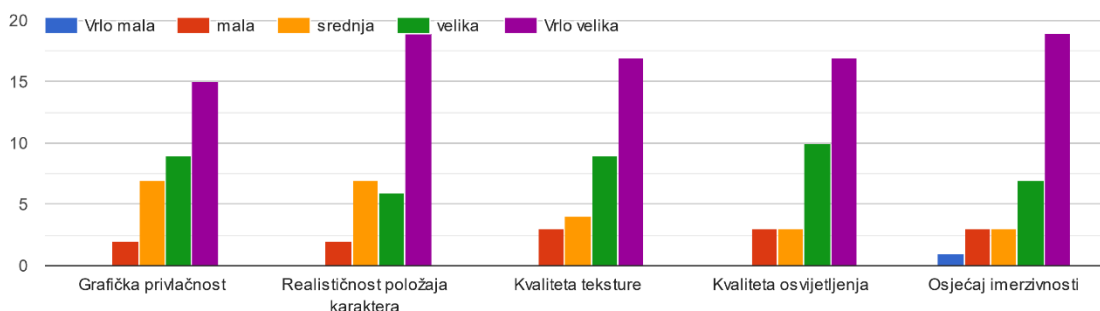
Za treću sliku (Slika 70), 15 ispitanika je ocijenilo „Grafičku privlačnost“ kao vrlo visoku, 9 kao visoku, 7 kao srednju i 2 kao nisku. Što se tiče „Realističnosti scene“, 19 ispitanika je ocijenilo kvalitetu kao vrlo visoku, 6 kao visoku, 7 kao srednju i 2 kao nisku. Za „Kvalitetu tekstura“, 17 ispitanika je ocijenilo kvalitetu kao vrlo visoku, 9 kao visoku, 4 kao srednju i 3 kao nisku. „Kvalitetu osvjetljenja“, 17 ispitanika je ocijenilo kvalitetu kao vrlo visoku, 10 kao visoku, 3 kao srednju i 3 kao nisku. Za „Osjećaj imerzivnosti“, 19 ispitanika je ocijenilo kvalitetu kao vrlo visoku, 7 kao visoku, 3 kao srednju, 3 kao nisku i 1 kao vrlo nisku (Slika 71).



Slika 70: Anketa – slika 3 iz ankete

Izvor: Autor

Odgovorite na sljedeća pitanja o slici 3. davanjem ocjene od vrlo mala, do vrlo velika za svaki ponuđeni opis



Slika 71: Anketa – ocjene slike 3 iz rada

Izvor: Autor

3.5. RASPRAVA REZULTATA

Budući da se radi o tehnologiji i tržištu koje je postalo izrazito popularno među širom publikom tek u posljednjih deset godina, bilo je važno istražiti koja dobna skupina prevladava u istraživanju. Osim toga, trebalo je uzeti u obzir obrazovnu pozadinu jer se radi o specifičnoj tehnologiji koja je više naglašena u određenim područjima, kao što su tehničke, tehnološke i umjetničke discipline. Ovo sugerira da osobe s obrazovanjem iz tih područja vjerojatno imaju dublje razumijevanje tematike koja se istražuje u anketi i, prirodno, su više upućene u ovakvu vrstu tehnologije.

Iz ukupnog broja sudionika koji su ispunjavali anketu, njih 60,6% je završilo višu razinu obrazovanja, a značajan broj njih potječe iz tehničkog i informatičkog područja, što sugerira da većina ispitanika koristi računalo svakodnevno i vjerojatno ima barem osnovno razumijevanje Unreal Engine-a i njegove svrhe.

Podaci su također pokazali da i izvan ovog tehničkog kruga postoje brojni korisnici računala, s čak 100% ispitanika koji su izjavili da koriste računalo svaki tjedan. Od tog broja, čak 81,1% koristi računalo svakodnevno, dok 12,1% koristi računalo nekoliko puta tjedno.

Ispitanicima je postavljeno i pitanje o učestalosti igranja video igara, pri čemu je čak 81,8% ispitanika izjavilo da igra video igre svaki tjedan. Ovo ukazuje na to da ispitanici često koriste video igre kao svoj glavni medij zabave. Ova činjenica nije iznenađujuća s obzirom na to da je veći dio ispitanika u dobi između 18 i 30 godina, što je dobna skupina koja često konzumira ovakav tip zabave.

Iznenađujuće je da čak 97% ispitanika je čulo za pojam „Game Engine“, a 69,7% njih zna otprilike ili točno čemu „Game Engine“ služi. Ovo ukazuje na to da čak i ispitanici koji nisu stručnjaci u području informatike i tehnologije su upoznati s ovim programima i da dio njih razumije njihovu svrhu, iako njihova struka nije povezana s ovim tehnologijama.

Niz sljedećih pitanja bio je fokusiran na grafička svojstva video igara s ciljem utvrđivanja važnosti grafike u igri i njenog utjecaja na potencijalnu kupnju ispitanika. Pokazalo se da veliki broj ispitanika smatra grafiku važnim aspektom pri odabiru igre. Čak 90,9% ispitanika je izjavilo da im je grafika bitna ili bitna, ali nije glavni prioritet.

Osim toga, važnost grafike u video igrama se pokazala i kroz njihov odnos prema cijeni igre. Visokih 84,8% ispitanika je izjavilo da im je grafika bitna prilikom odlučivanja o kupnji video igre. Zanimljiv podatak je da velik broj ispitanika izražava spremnost da plati više za igru koja nudi izvanrednu grafiku u usporedbi s igrama koje su vizualno manje impresivne. Ovo sugerira da igre koje stavljaju naglasak na grafičko oblikovanje potencijalno mogu postaviti višu cijenu za svoj proizvod u usporedbi s igrama gdje grafika nije primarni fokus dizajna. Nadalje, čak 90,7% ispitanika smatra da grafika u igrama može značajno poboljšati cjelokupno iskustvo igranja.

Kako bismo saznali kakav tip grafike ispitanici više preferiraju, postavili smo pitanje o njihovim preferencijama u vezi s grafikom u igrama, pitajući ih da li više favoriziraju realističnu ili stiliziranu grafiku. Rezultati su pokazali da čak 71,9% ispitanika preferira realistični stil grafike, dok 15,6% nije sigurno u svoje preferencije ili nemaju izraženi stil koji preferiraju. Ovi rezultati sugeriraju da kvalitetno izvedena realistična grafika ostavlja jači dojam na ispitanike i potencijalno može biti ključni faktor za njihovu odluku o kupnji video igre. Ovo je u skladu s prethodnim odgovorima

ispitanika gdje su istaknuli važnost grafike pri odabiru igre, što dodatno potvrđuje da kvaliteta grafike može značajno utjecati na njihovu odluku o kupnji.

Kada je riječ o grafičkim svojstvima igre, istraživanje je pokazalo da sva vrsta grafičkih karakteristika igraju važnu ulogu u stvaranju dojma visokokvalitetnog grafičkog rješenja. Istraživanje je pokazalo da između animacije, kvalitete teksture i kvalitete osvjetljenja, svaki od ovih aspekata čini ključni dio cjeline koja je važna većini ispitanika kako bi grafičko rješenje bilo funkcionalno i ostvarilo visoku razinu kvalitete.

Završni dio istraživanja uključivao je prikaz tri slike koje predstavljaju konačno grafičko rješenje igre s fokusom na realističan prikaz grafike. S obzirom na prethodne odgovore ispitanika o njihovim preferencijama u pogledu vrste grafike i utjecaju grafike na ukupni dojam, ove slike su podvrgnute sličnom ocjenjivanju.

U većini slučajeva, ocjene ponuđenih slika su ukazale na vrlo visoku ili visoku kvalitetu prikaza, uz nekoliko ocjena koje su označene kao srednje ili niske kvalitete grafičke privlačnosti. Ovo sugerira da je predstavljeno grafičko rješenje kvalitetne prirode, ali i da postoji uvijek prostor za poboljšanja i nadogradnje, što je uobičajeno u razvoju grafičkih sustava. Realističan prikaz grafike nikada nije potpuno savršen, a svaka nova iteracija donosi napredak u kvaliteti.

Istraživanje je također pokazalo da grafički elementi i grafička rješenja imaju značajan utjecaj na kvalitetu proizvoda, a potencijalno i na njegovu cijenu. Spremnost potrošača da plate više za proizvod često ovisi o njegovom vizualnom izgledu, što je usko povezano s grafičkim rješenjem proizvoda. Osim toga, kako bi se postigao pravi realističan prikaz, potrebna je visoka kvaliteta svih aspekata grafike koji zajedno čine cjelinu.

4. ZAKLJUČAK

Alati za renderiranje interaktivne računalne grafike u stvarnom vremenu danas su vrlo moćna dizajnerska i programerska sredstva. Njihova sposobnost stvaranja fotorealističnih vizualnih i interaktivnih iskustava čine ih ključnim alatima za razvoj modernih igara. S obzirom na kontinuirani razvoj tehnologija kao što su virtualna stvarnost i proširena stvarnost, ovi alati će imati značajnu ulogu u budućnosti dizajna.

Grafički sustavi i tehnologije Unreal Engine-a sa svakom inačicom pomiču standarde industrije. Inovativne tehnologije kao „Nanite“ i „Lumen“ mijenjaju pristupe i mogućnosti dizajna interaktivne grafike. Ove inovacije su omogućile dizajnerima da nadvladaju izazove te stvore kompleksna okruženja koja bi bila nedostižna koristeći starije alate. Proceduralni elementi engine-a uvelike ubrzavaju način rada i omogućavaju timovima limitiranih resursa puno efikasniji model dizajna, što obogaćuje tržište kreativnom snagom pojedinaca, te potencijalno pruža inovativnost koja u prijašnjim alatima nije bila u mogućnosti biti eksponirana na današnjim razinama.

Alati kojima je prvobitna svrha produkcija kreativnosti imaju snažne temelje za promjenu i inovaciju, te pomicanju i građenju novih granica. Takvi alati predstavljaju skupinu koja pruža brze i efikasne metode, te ono najbitnije u takvome sustavu, širok spektar primjene, te impresivna vizualna iskustva.

5. LITERATURA

- [1] Moniem MA. 2015. *Unreal Engine Lighting and Rendering Essentials*. Packt Publishing.
<http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1071003>. Pristupljeno Kolovoz 20 2023.
- [2] Patrik Lazić, Tibor Skala, Marko Maričević, Lara Maričević. 2023. *Renderiranje interaktivne računalne grafike u stvarnom vremenu za potrebe razvoja kompjuterskih igara*. J. Ž. Gršić, Printing&Design 2023.
<https://www.tiskarstvo.net/printing&design2023/ZbornikRadovaTISKARSTVO2023.pdf> Pristupljeno Rujan 19 2023.
- [3] Tomas Akenine-Mller, Eric Haines, and Naty Hoffman. 2018. *Real-Time Rendering, Fourth Edition (4th. ed.)*. A. K. Peters, Ltd., USA. Pristupljeno Kolovoz 24 2023.
- [4] Rendering in Computer Graphics, <https://www.tutorialandexample.com/rendering-in-computer-graphics#> Pristupljeno Kolovoz 27 2023.
- [5] Real-Time Rendering Resources <https://www.realtimerendering.com/> Pristupljeno Kolovoz 27 2023.
- [6] Gregory, J. (2018). *Game engine architecture*. crc Press Pristupljeno Kolovoz 24 2023.
- [7] Introduction to Blueprints, Unreal Engine Document Page,
<https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/GettingStarted/> Pristupljeno Kolovoz 27 2023.
- [8] Blueprints Visual Scripting, Unreal Engine Document Page,
<https://docs.unrealengine.com/5.1/en-US/blueprints-visual-scripting-in-unreal-engine/> Pristupljeno Kolovoz 27 2023.

- [9] Blueprint Variables: What you need to know, <https://www.unrealdirective.com/articles/blueprint-variables-what-you-need-to-know>
Pristupljeno Kolovoz 27 2023.
- [10] Blueprint Variables, Unreal Engine Document Page, <https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Blueprints/UserGuide/Variables/> Pristupljeno Kolovoz 27 2023.
- [11] Creating Landscapes, Unreal Engine Document Page, <https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/Landscape/Creation/>
Pristupljeno Kolovoz 24 2023.
- [12] Foliage Tool, Unreal Engine Document Page, <https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/Foliage/> Pristupljeno Kolovoz 24 2023.
- [13] Working with UV Channels, <https://docs.unrealengine.com/4.26/en-US/WorkingWithContent/Types/StaticMeshes/HowTo/UVChannels/> Pristupljeno Kolovoz 24 2023.
- [14] Organizing a Material Graph, Unreal Engine Document Page, <https://docs.unrealengine.com/5.2/en-US/organizing-a-material-graph-in-unreal-engine/>
Pristupljeno Kolovoz 25 2023.
- [15] Material Parameter Collections, <https://www.unrealengine.com/en-US/blog/material-parameter-collections> Pristupljeno Kolovoz 25 2023.
- [16] Texture Expressions, Unreal Engine Document Page, <https://docs.unrealengine.com/4.26/en-US/RenderingAndGraphics/Materials/ExpressionReference/Textures/> Pristupljeno Kolovoz 25 2023.
- [17] Material Parameter Expressions, Unreal Engine Document Page, <https://docs.unrealengine.com/5.2/en-US/material-parameter-expressions-in-unreal-engine/> Pristupljeno Kolovoz 26 2023.

[18] Shader Development, Unreal Engine Document Page,
<https://docs.unrealengine.com/4.26/en-US/ProgrammingAndScripting/Rendering/ShaderDevelopment/> Pristupljeno Kolovoz 27 2023.

[19] Nanite Virtualized Geometry, Unreal Engine Document Page,
<https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine/> Pristupljeno Srpanj 14 2023.

[20] Global Illumination, Unreal Engine Document Page
<https://docs.unrealengine.com/4.27/en-US/RenderingAndGraphics/GlobalIllumination/> Pristupljeno Srpanj 14 2023.

[21] Lumen Global Illumination and Reflections, Unreal Engine Document Page
<https://docs.unrealengine.com/5.0/en-US/lumen-global-illumination-and-reflections-in-unreal-engine/> Pristupljeno Srpanj 14 2023.

6. POPIS SLIKA

Slika 1: Početni prozor Unreal editora

Slika 2: Scanline algoritam

Slika 3: Građenje slike sa „Ray-tracing“ algoritmom

Slika 4: Koraci procesa „Radiosity“

Slika 5: Četiri faze toka renderiranja u stvarnom vremena

Slika 6: Primjer Blueprinta

Slika 7: Pejzažni sustav

Slika 8: Primjer modifikacije pejzaža s alatom Oblikovanje

Slika 9: Primjer modifikacije pejzaža s alatom Oblikovanje

Slika 10: Pejzaž cijelog nivoa, prva perspektiva

Slika 11: Pejzaž cijelog nivoa, druga perspektiva

Slika 12: Uporaba „Foliage“ kista

Slika 13: Uporaba „Foliage“ kista

Slika 14: „Foliage“ trave korišten u radu

Slika 15: „Foliage“ biljke maline korišten u radu

Slika 16: Promjena parametara dinamičkog auto materijala

Slika 17: Promjena parametara dinamičkog auto materijala

Slika 18: Parametri dinamičkog auto materijala

Slika 19: „Blueprint“ sistem, Grafikon čvorova

Slika 20: Normal mapa korištenog modela u radu

Slika 21: Blueprint „layera“ trava (eng. Grass)

Slika 22: „Blueprint“ DepthFadeMask_R

Slika 23: „Blueprint“ Maske za teksture

Slike 24: Primjer 1. promjene parametara vode u dinamičnom materijalu

Slike 25: Primjer 2. promjene parametara vode u dinamičnom materijalu

Slike 26: Primjer 3. promjene parametara vode u dinamičnom materijalu

Slike 27: Primjer 4. promjene parametara vode u dinamičnom materijalu

Slika 28: „Blueprint“ za ispisivanje parametara vode u dinamičnom materijalu

Slika 29: Vizualni sadržaj igre

Slika 30: Prikaz „Nanite“ tehnologije kroz poligone u vizualnom sadržaju igre

Slika 31: Primjer 1. Promjena osvjetljenja u stvarnome vremenu

Slika 32: Primjer 2. Promjena osvjetljenja u stvarnome vremenu

Slika 33: Položaj kamere za animaciju igrača

Slika 34: Niz pripremljenih animacija za kombiniranje

Slika 35: Komponiranje niza animaciju u jednu „miješanu animaciju“

Slika 36: Gotov primjer animacije tijela iz perspektive igrača

Slika 37: Gotov primjer animacije tijela u pokretu iz perspektive igrača

Slika 38: Primjer 1. animacija nošenja oružja iz perspektive igrača

Slika 39: Primjer 2. animacija nošenja oružja iz perspektive igrača

Slika 40: Primjer 1. Pejzaž nivoa

Slika 41: Primjer 2. Pejzaž nivoa prekriven snijegom

Slika 42: Primjer 3. Pejzaž nivoa

Slika 43: Primjer 4. Pejzaž nivoa

Slika 44: Prvi pomak od četiri pomaka za animaciju

Slika 45: Drugi pomak od četiri pomaka za animaciju

Slika 46: Treći pomak od četiri pomaka za animaciju

Slika 47: Četvrti pomak od četiri pomaka za animaciju

Slika 48: Karakter u pokretu

Slika 49: Primjer 5. Pejzaž nivoa

Slika 50: Primjer 6. Pejzaž nivoa

Slika 51: Anketa - Spol

Slika 52: Anketa - Dob

Slika 53: Anketa – Razina obrazovanja

Slika 54: Anketa – Područje obrazovanja

Slika 55: Anketa – Učestalost korištenja računala

Slika 56: Anketa – Konzumacija igrice

Slika 57: Anketa – Učestalost igranja igrice

Slika 58: Anketa – Poznavanje pojma „Game engine“ pitanje 1.

Slika 59: Anketa – Poznavanje pojma „Game engine“ pitanje 2.

Slika 60: Anketa – Važnost grafike u izboru igre

Slika 61: Anketa – Važnost grafike u odnosu na cijenu igrice

Slika 62: Anketa – Važnost grafike u odnosu na cijenu igrice

Slika 63: Anketa – Važnost grafike u odnosu na cijenu igrice

Slika 64: Anketa – Preferiranje tipa grafike

Slika 65: Anketa – Što privlači najveću pažnju u grafici

Slika 66: Anketa – slika 1 iz ankete

Slika 67: Anketa – ocjene slike 1 iz rada

Slika 68: Anketa – slika 2 iz ankete

Slika 69: Anketa – ocjene slike 2 iz rada

Slika 70: Anketa – slika 3 iz ankete

Slika 71: Anketa – ocjene slike 3 iz rada

7. POPIS MANJE POZNATIH RIJEČI

Engine – softversko okruženje za razvoj videoigara

Asseti - Bilo koji dio sadržaja u Unreal Engine-u (oblici, materijali,)

Frame rate - mjerenje brzine kojom se broj sličica pojavljuje u jednoj sekundi

Hertz - standardna jedinica frekvencije u Međunarodnom sustavu jedinica (SI)

RGB vrijednosti - označava crveni, zeleni i plavi intenzitet boje

NVIDIA - Američka multinacionalna tehnološka tvrtka

Mesh - Osnovna jedinica koja se koristi za stvaranje geometrije nivoa

Pre-render – unaprijed renderani sustav

Rasterizacija – proces uzimanja slike opisane u vektorskom grafičkom formatu i pretvaranja u rastersku sliku

Fotorealistično - Stvarni prikaz scene

Piksel - najmanji grafički element bitmap slike

Layer - različite razine na koje se postavljaju objekti

Poligon – višekut, zatvoreni geometrijski lik omeđen dužinama (trokut)

Proceduralan - zapisivanje popisa uputa koje će računaru reći što treba učiniti

Maska - tekstura u sivim tonovima ili jedan kanal teksture, koji se koristi za ograničavanje područja efekta unutar materijala

Foliage - kolekcija oblika poput trave, lišća i cvijeća