



KOMPARATIVNA ANALIZA RENDERER-A U BLENDER-U: USPOSTAVLJANJE RAVNOTEŽE IZMEĐU BRZINE I FOTOREALIZMA

COMPARATIVE ANALYSIS OF BLENDER RENDER ENGINES: THE BALANCE OF SPEED AND PHOTOREALISM

Jelena Šušnjević, Fakultet tehničkih nauka, Novi Sad

Oblast - RAČUNARSKA GRAFIKA

**Kratak sadržaj** — Ovaj rad poredi dva render-era u Blender-u, Eevee i Cycles, analizirajući njihove nedostatke i prednosti, kad su u pitanju brzina, realizam i efikasno korišćenje kompjuterskih resursa. Procenjuje različite upotrebe za oba render-era, pružajući uvid u različitosti između rendering-a u realnom vremenu i produkcije realistične 3D slike. Stečeno znanje je iskorišćeno u praktičnom delu rada, za rendering specifične 3D scene.

**Ključne reči:** render, algoritam, onlajn, oflajn

**Abstract** — This paper compares two render engines within Blender, Eevee, and Cycles, analyzing their weaknesses and strengths in terms of speed, realism and resource utilization. It evaluates different uses for both render engines, providing insights into the trade-offs between real-time rendering, and photorealistic image production. The acquired knowledge was used in the practical part of the work to render a specific outdoor scene.

**Keywords:** render, algorithm, online, offline

1. UVOD

U svetu 3D kompjuterske grafike koja se konstantno razvija, izbor renderera je od ključne važnosti, jer u zavisnosti od toga da li dajemo prioritet brzini renderovanja ili realizmu, možemo da biramo između dva tipa render-era, online i offline. Blender, open source program za 3D modelovanje, nudi dva istaknuta renderera, Eevee i Cycles, koji su namenjeni da se koriste u različitim uslovima i prilagođeni su različitim potrebama.

Cilj ovog rada jeste istraživanje i komparativna analiza renderera Eevee i Cycles u Blender-u, sa fokusom na njihove prednosti, slabosti i balans između brzine renderovanja i fotorealističnog rezultata. Kako potražnja za mogućnostima renderovanja u realnom vremenu nastavlja da raste, Eevee se izdvaja kao odličan izbor, koji omogućava izradu brze i interaktivne vizuelizacije.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Ratko Obradović, red. prof.

S druge strane, render-er Cycles, poznat po svojoj *path-tracing* karakteristici, ima za cilj da pruži odličan realizam, ali po cenu sporijeg render-ovanja.

2. TEORIJSKI DEO

U ovom poglavlju analiziramo kompleksnost procesa ovih render-era, proučavajući algoritme na osnovu kojih funkcionišu. Počevši od mogućnosti render-ovanja u realnom vremenu, skoro trenutno nakon svake, promene ili pomeranja, pa do render-ovanja offline, imitiranjem ponašanja svetlosti, posmatračemo različitosti ova dva tipa render-ovanja i doneti zaključak kada treba koji koristiti, u kojim situacijama možemo da iskoristimo njihove prednosti i kako da nadomestimo njihove mane. Ovo poglavlje ima za cilj da otkrije različite karakteristike koje definišu princip rada ova dva render-era.

Bavićemo se praktičnim primenama Eevee i Cycles render-era, u kakvim kreativnim scenarijima treba da se opredelimo za jedan od njih, ili ukoliko koristimo oba, u kom delu *pipeline*-a rada treba da koristimo koji. Izučavaćemo njihove individualne potencijale, praveći kompromis između brzine, vizuelne vernosti i potrebnim računarskim resursima. Bilo da se radi o optimizaciji za kreiranje interaktivnog sadržaja, arhitektonskoj vizuelizaciji ili produkciji animiranog filma, razumevanje razlika između Eevee-a i Cycles-a je neophodno za donošenje odluka kad dođe do njihove upotrebe u različitim 3D projektima.

Render-ovanje u realnom vremenu uključuje pojam interaktivnosti, jer se scena, sastavljena od mnoštva slika, dobije trenutno, neposredno pre nego što se prikaže na ekranu. Ukratko, računar proizvodi sve rendere iz 3D geometrije, teksture i svetla i prikazuje ih korisniku što je brže moguće. Ovaj tip render-era je veoma zastupljen u svetu digitalnih medija, ali najbolji primer njegove upotrebe jeste industrija video igara. Programeri i umetnici, u industriji video igara, imaju izazov da naprave što lepšu i realističniju grafiku, koja će se prikazivati u realnom vremenu. Korisnici, intereaguju sa scenom, kreću se kroz virtuelni prostor i svaka izmena koju naprave treba da se prikaže što brže na ekranu, inače gubimo imersivnost.

Eevee je napravljen korišćenjem *OpenGL*-a i fokusira se na brzinu i interaktivnost, dok ima mogućnost da renderuje PBR (eng. Physically Based Rendering) materijale. Eevee koristi proces koji se zove rasterizacija, procenjuje način na koji svetlost intereaguje sa objektima

i materijalima koristeći brojne algoritme. Iz tog razloga, ima veliki set ograničenja.

Neka ograničenja uključuju:

krvarenje svetlosti

mape senki niske rezolucije

nedostatak CPU opcije

ograničenu ambijentalnu okluziju

efekat globalne osvetljenosti

refleksiju

transparentost

refrakciju

Većina ovih nedostataka je proizvod Eevee renderer-ovog brzog vremena render-ovanja. Zato što Eevee ne funkcioniše na istom principu kao Cycles, simuliranjem ponašanja svetlosti, već raznim algoritmima “vara” i nagađa koji objekti bi mogli biti osenčeni, vidljivi, prozirni,...

Prednosti Eevee renderer-a jesu brzina i pametno korišćenje kompjuterskih resursa. Još jedna velika prednost Eevee render-a, je ta da se ne mora birati između njega i Cycles-a, već mogu da se koriste kombinovano, u različitim fazama rada. Zbog svoje velike brzine render-ovanja, pored kreiranja finalnih render-a za video igre, moguće je kreirati PBR materijale sa neverovatnom preciznošću, pregledanje animacije istog momenta i kreiranje game asset-a. Ove prednosti omogućavaju korisnicima da, pre nego da pokrenu finalan render koji može trajati danima, naprave nešto što je približno finalnoj verziji. U tom slučaju, Eevee se koristi kao prerender, dok se Cycles koristi kao za kreiranje finalnog rendera.

Offline render-ovanje odnosi se na situaciju gde se frejmovi prikazuju, ne u realnom vremenu, već nakon kraćeg ili dužeg vremenskog perioda. Iako je vreme render-ovanja znatno kraće nego kod online render-era, offline render-er proizvodi veoma kvalitetnu grafiku koja je bliska stvarnosti. Cycles je Blender-ov *rendering* sistem zasnovan na imitiranju ponašanja svetlosti korišćenjem fizike. Dizajniran je da obezbedi realistične rezultate, sa umetničkom kontrolom i fleksibilnošću za produkcijske potrebe. Cycles koristi *path tracing* algoritam da prati putanje svetlosnih zraka, omogućavajući vernu reprodukciju složenih svetlosnih fenomena, kao što su refleksija, prelamanje svetlosti i globalno osvetljenje. Ono što izdvaja Cycles jeste njegova posvećenost pridržavanja zakonima fizike kako bi se precizno simuliralo ponašanje svetlosti.

Ograničenja Cycles render-era najviše su vezana za brzinu render-ovanja. Obzirom na to da Cycles troši mnogo više resursa kako bi proizveo verodostojnu sliku, treba mu mnogo više vremena da izvrši sve svoje kalkulacije. Vreme render-ovanja može da se skрати, i računarski resursi mogu pametno da se iskoriste, ali čak i tada ne može da se poredi sa brzinom *realtime* render-era. Pored ograničenja brzine, najveće ograničenje *Cycles renderer-a* jeste to što ima maksimalan broj odbijanja svetlosti. U prirodi, svetlosni zrak se odbija beskonačan broj puta, dok se svetlost potpuno ne apsorbuje od strane predmeta sa kojima intereaguje, dok je taj broj kod 3D

render-a ograničen. Još jedno značajno ograničenje jeste loša kaustika, iako je fokus *Cycles renderer-a* stvaranje realističnih slika, kad dođe do kaustike, ne proizvodi rezultate koji mogu da se poredi sa drugim rendererima.

Neke prednosti Cycles render-era su:

- Realistični materijali
- GPU rendering
- PBR
- Globalna osvetljenost
- Adaptive Sampling.

### 3. PRAKTIČNI DEO

Ovo poglavlje ima za cilj da opiše modelovanje scene, postavljane tekstura i osvetljenja, a naglasak je na podešavanjima render-era tako da dobijemo što bolje rezultate, uzimajući u obzir njihova ograničenja.

Prvo je kreirana 3D scena u Blender-u, pasaż koji je za inspiraciju imao postojeći pasaż u Zmaj Jovinoj ulici.



Slika 1 Modelovana scena bez tekstura i materijala

Neki objekti su modelovani dok su drugi samo uvezeni iz besplatne biblioteke kao što je *Megascans*. Materijali su pravljani ili povezivanjem *node-ova*, ili su uvezeni iz *asset library* “*Crafty asset pack*”. Neki materijali su preuzeti sa interneta i uvezeni uz omogućenu opciju *Node Wrangler* koja znatno olakšava proces rada, jer ukoliko preuzmemo teksturu sa interneta, potrebno je samo da napravimo običan *principal bsdf shader* i na njega automatski primenimo sve mape teksture koje smo uvezli.

Pošto Eevee *render engine* ne može da koristi osvetljenje iz *HDRI* mape, moramo da lažiramo to osvetljenje. Ubacujemo *sun light* i pozicioniramo ga otprilike gde bi se zapravo nalazilo Sunce. Zatim, moramo da lažiramo svetlost neba, pošto svi objekti koji nisu u kontaktu direktno sa *sun light* će biti crni. To postizemo ubacivanjem *area light*, koji pozicioniramo direktno iznad scene i povećamo. Pošto Eevee *render engine* ne kalkuliše globalnu osvetljenost, to jest ne računa indirektnu svetlost, ubacujemo *irradiance volumes probe*. *Irradiance volumes* se odnosi na tehniku koja se koristi za simulaciju efekta globalnog osvetljenja u sceni. Ovo uključuje prethodno izračunavanje ili čuvanje informacija o distribuciji svetlosti unutar prostora, omogućavajući realističnije efekte indirektnog osvetljenja.



Slika 2 Postavka lažnog osvetljenja

Kod Eevee renderer-a, bitno je da isključimo HDRI mapu kao uticaj na osvetljenje, jer neće stvarati realistične senke, već samo uticati negativno na finalan rezultat. Kod Eevee render-era, da bismo dobili realističniju sliku, možemo da omogućimo dodatna podešavanja.

Bloom je efekat koji simulira raspršivanje svetlosti oko svetlih objekata u sceni. Kada je omogućen, poboljšava percipiranu osvetljenost i sjaj izvora svetlosti visokog intenziteta.

Iako Eevee ne podržava raspršivanje svetlosti, ima pojednostavljenu opciju sličnog efekta. Oslanja se na informacije dostupne u prozoru koji vidimo, što znači da možda neće uhvatiti sve detalje prodiranja svetlosti i rasipanja svetlosti ispod površine.

Screen Space Reflections (SSR) je opcija koja se koristi za simulaciju realističnih refleksija u sceni. SSR izračunava refleksije na osnovu onoga što je trenutno vidljivo na ekranu. SSR podešava refleksije kako se kamera ili objekti u sceni kreću, obezbeđujući vizuelno ubedljiv prikaz reflektujućih površina.

Takođe, Eevee ne može da simulira pojedine karakteristike svetlosti, pa će iz tog razloga proizvesti manje detaljne senke. Možemo ih poboljšati soft shadows ili light bleeding opcijom.

Što se tiče Cycles render-era, prvo podešavanje koje možemo uključiti jeste GPU compute. GPU se sastoji od hiljada manjih jezgara, što omogućava da istovremeno obavlja mnoge jednostavne zadatke, samo tom opcijom znatno ubrzavamo vreme render-ovanja.

Tiling opcija nam omogućava da umesto da render-ujemo celu sliku odjednom, render-ujemo više manjih delova koji čine jednu sliku. Ovo ne samo da ubrzava render, već je mnogo manje verovatno da će aplikacija prestati da radi.

Poslednje podešavanje koje možemo izmeniti jeste vezano za šum. Aktiviranjem noise threshold, dajemo Blender-u opciju da bira koliko će šuma otkloniti na kojim delovima slike.

Objekti sa manje detalja neće zahtevati istu količinu resursa koliko i objekti sa više detalja, na ovaj način smanjujemo naše vreme render-ovanja.

Globalna osvetljenost (eng. Fast GI) approximation menja globalno osvetljenje ambijentalnom okluzijom nakon određenog broja odbijanja. Ovo podešavanje može uticati negativno na realističnost render- a, ali može takođe znatno smanjiti vreme render-a.

Light paths podešavanje se odnosi na skup parametara koji kontrolišu kako svetlost stupa u kontakt sa

površinama i objektima u sceni tokom procesa render-ovanja. Ova podešavanja su posebno bitna za korisnike koji žele kontrolu nad načinom na koji se svetlost ponaša u scenama. Smanjivanjem broja odbijanja svetlosti, skraćuje se vreme renderovanja.

Još jedna opcija vezana za šum jeste denoise opcija, njenim aktiviranjem uklanjamo šum koji je nastao render-ovanjem. Pod denoiser stavljamo OpenImageDenoise, a pod passes stavljamo Albedo. OpenImageDenoise je biblioteka za uklanjanje šuma koja je dizajnirana da ubrza proces uklanjanja šuma sa render-a.



Slika 3 Eevee render



Slika 4 Cycles render

#### 4. ZAKLJUČAK

Nakon render-ovanja scene, možemo jasno da se vidimo da Cycles nudi mnogo realističniji rezultat. Sa druge strane, za svaki Cycles-ov render je bilo potrebno bar 7 minuta, dok je Eevee svoj render proizveo nakon par sekundi.

Primećeno je da Cycles mnogo bolje prikazuje biljke, to je zato što ima mogućnost da koristi prozirnost materijala, dok kod Eevee renderer-a to mora da se lažira. Takođe, osvetljenje scene je mnogo bolje kod Cycles-a, pogotovo indirektno osvetljenje.

Jasno se vidi i razlika u reflektivim materijalima, ova pojava je verovatno rezultat Screen Space Reflection Cube koji nije dobro izračunao koji objekti bi mogli biti reflektovani.

Poređenje između Blender-ovih Cycles i Eevee render engine-a otkriva dinamičku međusobnu igru brzine i fotorealizma. Cycles, koji funkcioniše na osnovu path tracing i physically based rendering, se ističe u pružanju cinemografskog realizma i preciznosti kroz svoj sistem node shader-a. S druge strane, Eevee, dizajniran za



renderovanje u realnom vremenu, daje prednost brzini i interaktivnosti za brze iteracije dizajna i interaktivna iskustva.

Izbor između Cycles i Eevee nije stroga odluka ili-ili, već zavisi od potreba projekta. Oni se međusobno dopunjuju unutar Blender ekosistema, nudeći raznovrsna rešenja za renderovanje.

## 5. LITERATURA

1. Cycles and Eevee: which renderer should we choose? [internet] - dostupno na:

<https://irendering.net/cycles-and-eevee-which-renderer-should-we-choose/>

2.Eevee essentials: Real Time Rendering in Blender [internet] - dostupno na:

<https://cgcookie.com/posts/eevee-essential-features-walkthrough-part-one>

5. Rendering with Blender's Eevee: Settings & Features [internet] - dostupno na:

<https://renderpool.net/blog/blender-eevee/>

6. How to render faster in Blender [internet] - dostupno na:

<https://www.vagon.io/blog/complete-guide-for-reducing-rendering-times-at-Blender/>

Svim adresama pristupljeno u oktobru i novembru 2023.

## 6. BIOGRAFIJA



**Jelena Šušnjević** rođena je 11.03.1998. godine u Novom Sadu, Republika Srbija. Završila je osnovnu školu "Petefi Šandor", a zatim gimnaziju "Svetozar Marković". Završila je osnovne akademske studije Animacija u inženjerstvu na Fakultetu tehničkih nauka 2021. godine. Radi kao animator unutar firme Barclays u Nju Jorku.