UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Škálovateľný proceduralny algoritmus pre trblietky v reálnom čase

Diplomová práca

Autor: Bc. Róbert Kica

Školiteľ: prof. RNDr. Roman Ďurikovič, PhD.

2024

Obsah

- Výsledky
- Úvod do problematiky

- Tvorba vlastného materiálu

- Existujúce metódy
- Vlastná metóda – vylepšenia a problémy
- Výskum

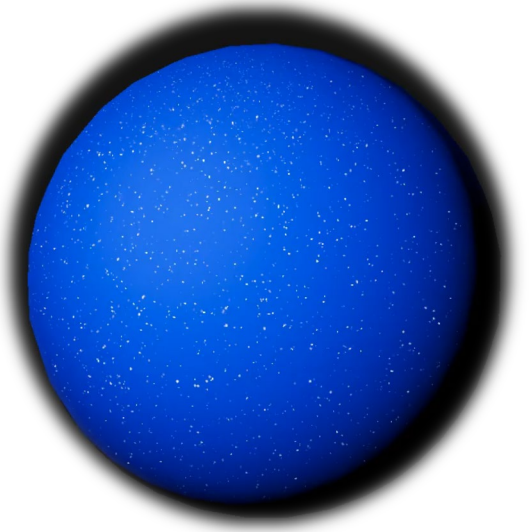
Ciele

- Navrhnúť a implementovať algoritmus vykresľovania trblietok v reálnom čase
- Vytvoriť materiál / Shader v Unity s rôznymi metódami, ktorý sa dá samostatne stiahnuť a použiť v iných Unity projektoch
- Porovnať, vyhodnotiť, jednotlivé metódy (vizuál, náročnosť a pod.)

Dosiahnuté výsledky

- Vytvorili sme **materiál** s rôznymi metódami trblietok pre Unity 2022.1 s HDRP 13.1.x.
  - Materiál sa dá stiahnuť + príloha: manuál na integráciu materiálu a nastavenie Unity
- Vytvorili sme **vlastnú metódu** – rozšírením metódy Beibei Wang a Huv Bowles.
  - Odstránili sme viditeľné pravidelné vzorkovanie
  - Hustejšie trblietky
- Vytvorili sme interaktívnu aplikáciu na porovnanie 4 známych metód a našej metódy
- Vyhodnotili sme jednotlivé metódy z vizuálneho a výkonnostného hľadiska
- S Unity + HDRP bolo niekedy ťažké pracovať (viď. sekcia 3.7 kapitoly Implementácia)

Jav trblietania

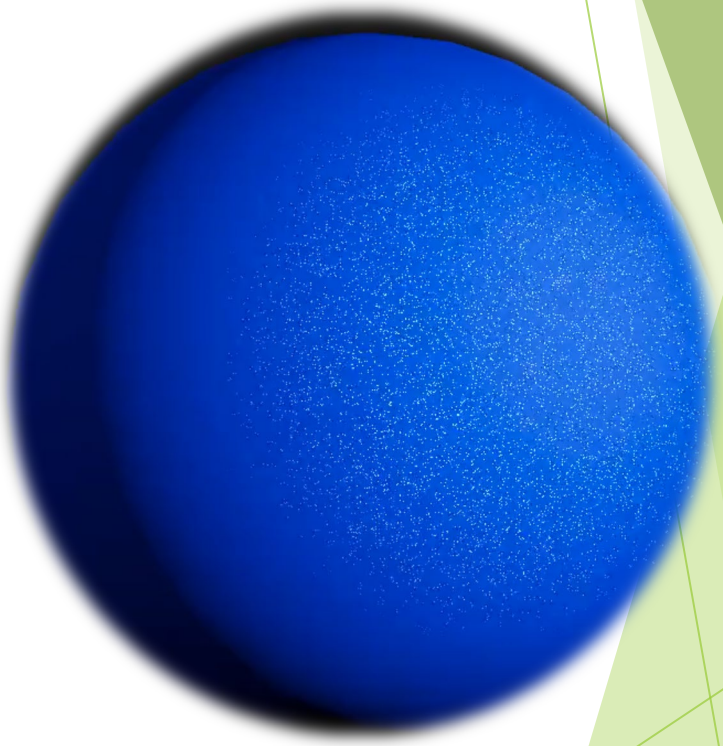


- Materiály majú zložitú mikroštruktúru
- Materiál s malými časticami na povrchu
- Častice v povrchu materiálu vytvárajú odrazy
  - Zrkadielka so silným odrazom

Zdroj: Recent advances in glinty appearance rendering ; Junqiu Z, Sizhe Z. et. al

Výzvy pri tvorbe trblietavého materiálu

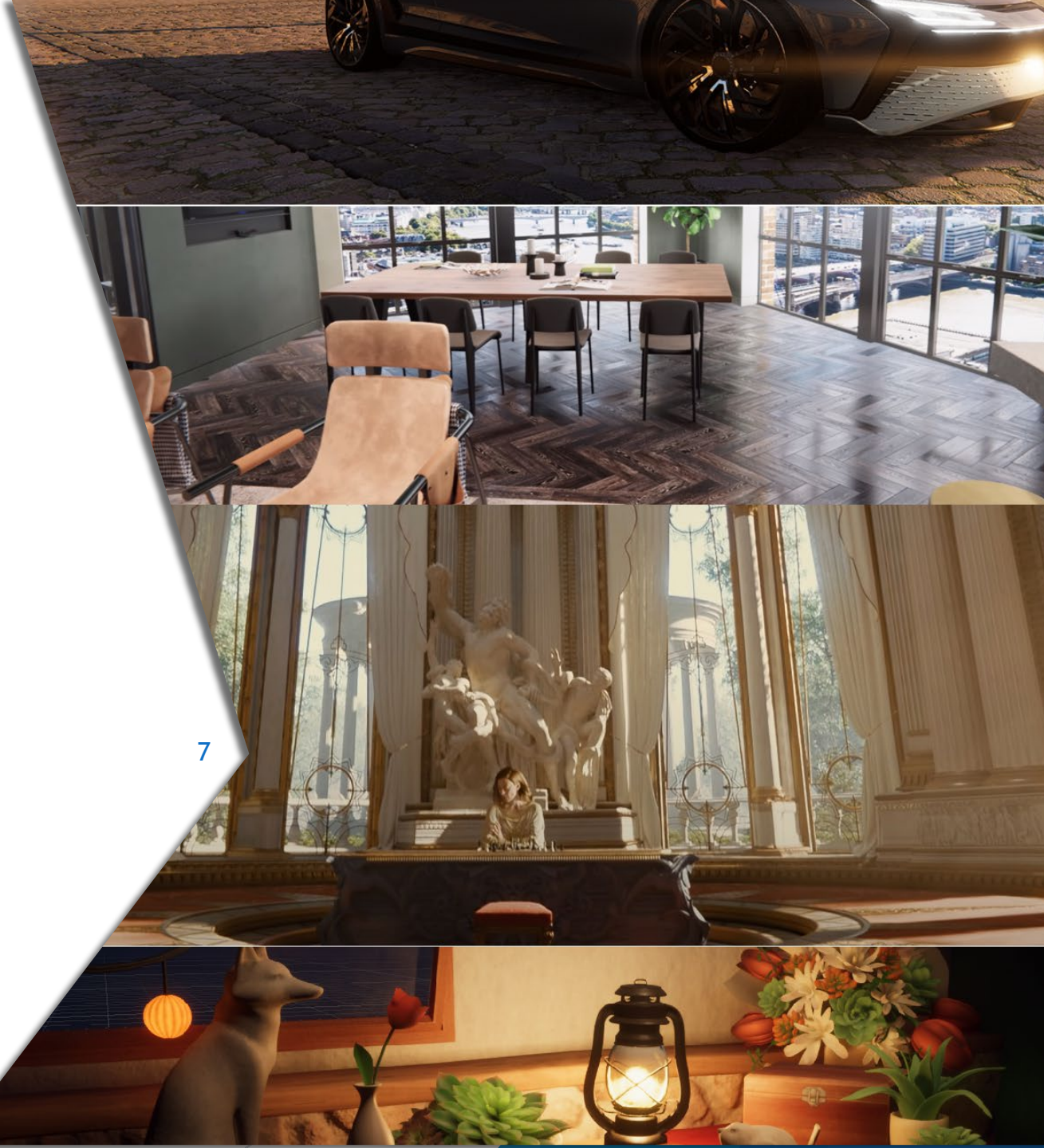
- Reprezentácia a vykresľovanie
  - Reprezentácia veľkého množstva malých častíc
  - Efektívne vykresľovanie
    - Najmä v reálnom čase – pri interaktívnom snímkovaní
- Koherencia, aliasing
- Hardvérová náročnosť, obmedzenia



Nežiadúce mihotanie

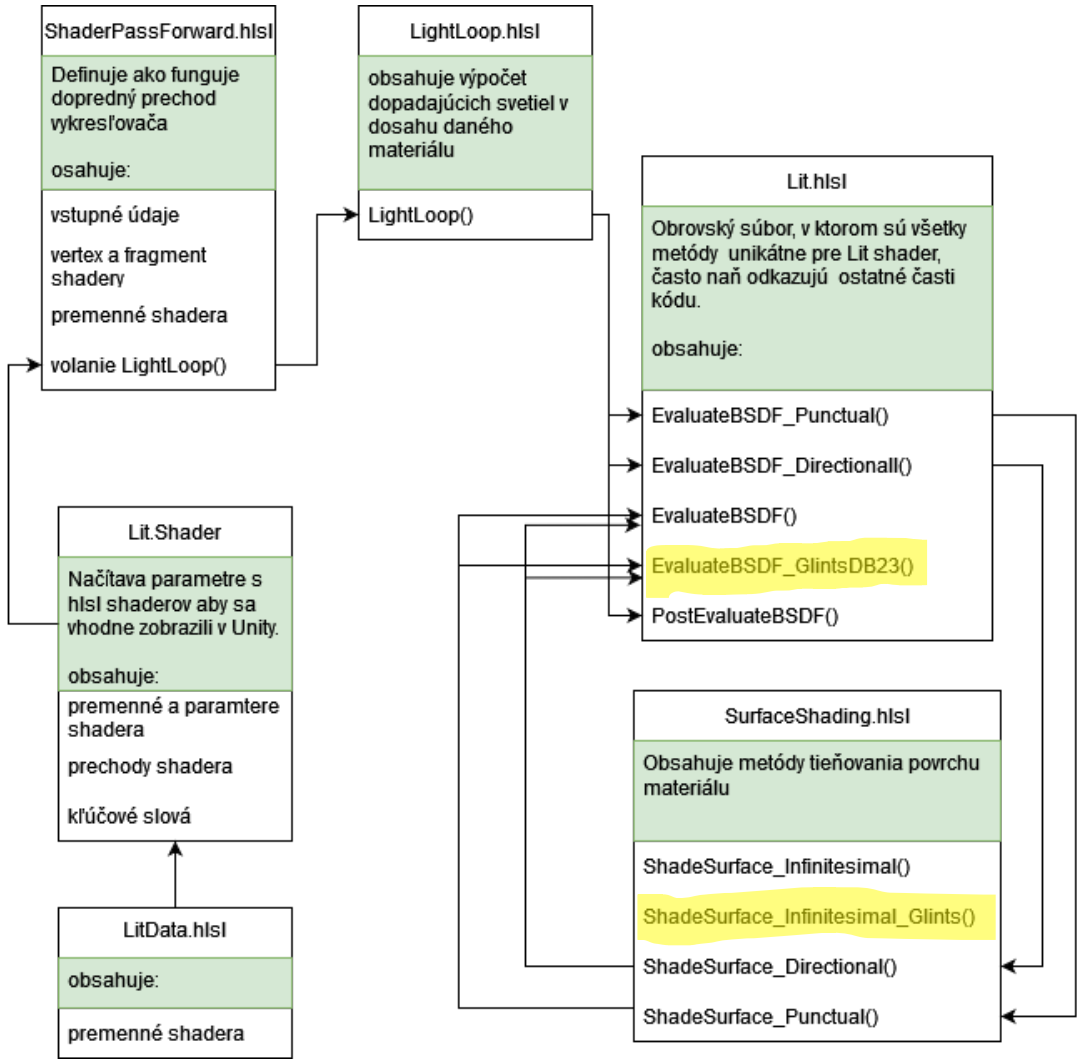
Vývojové prostredie

- Unity 3D 2022.1
  - primárne nástroj na tvorbu videohier menších štúdií, ale aj automobilovú vizualizáciu a filmový priemysel
- HDRP
  - Samostatný vykresľovací kanál pre výkonný hardvér s vysokou vizuálnou kvalitou
  - komplexný kanál
  - HLSL+ Unity makrá
  - horšia modifikácia



Zdroj: https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@17.0/manual/HDRP-Features.html

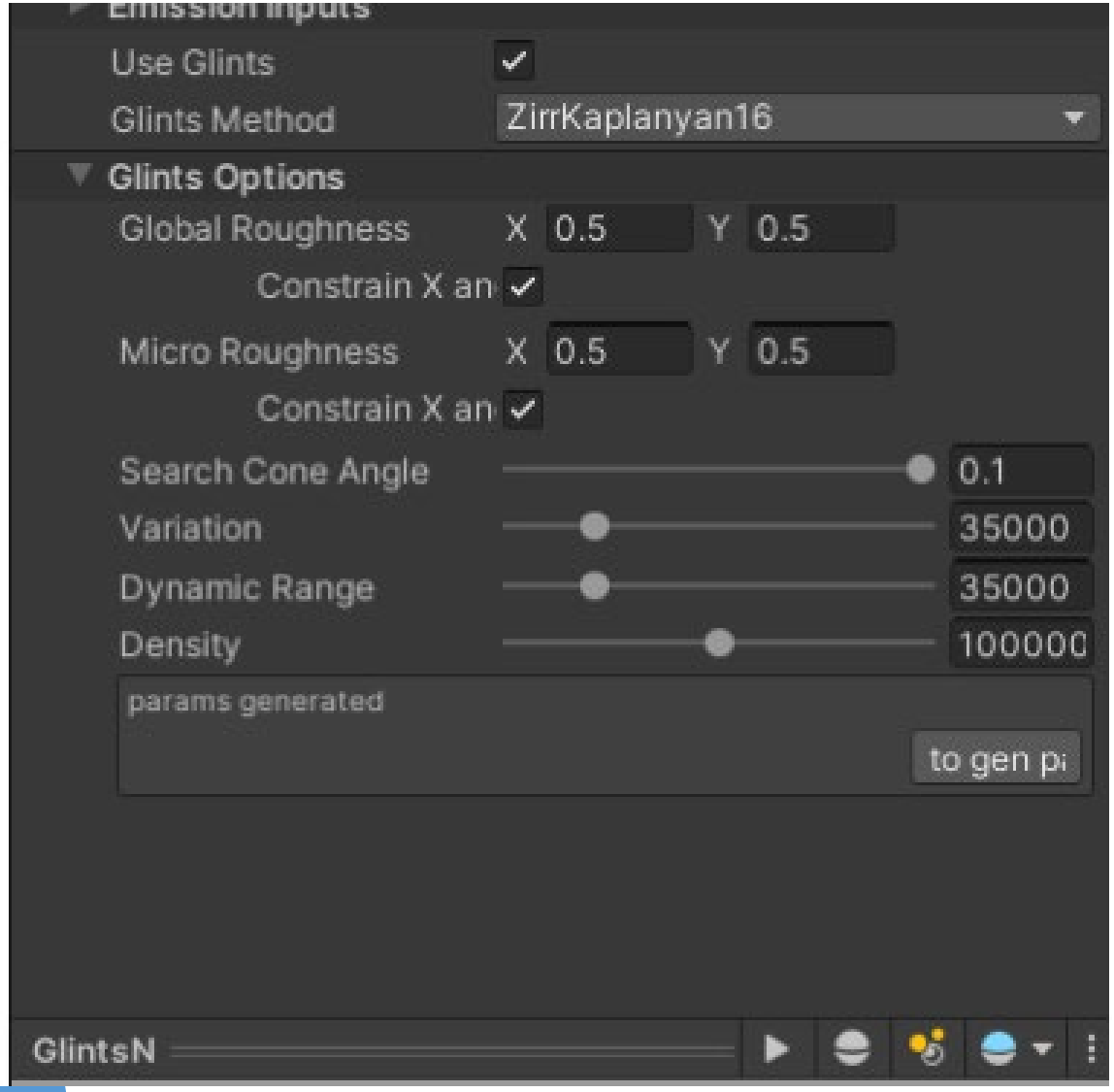
Znázornenie funkčnosti časti HDRP, ktorú modifikujeme

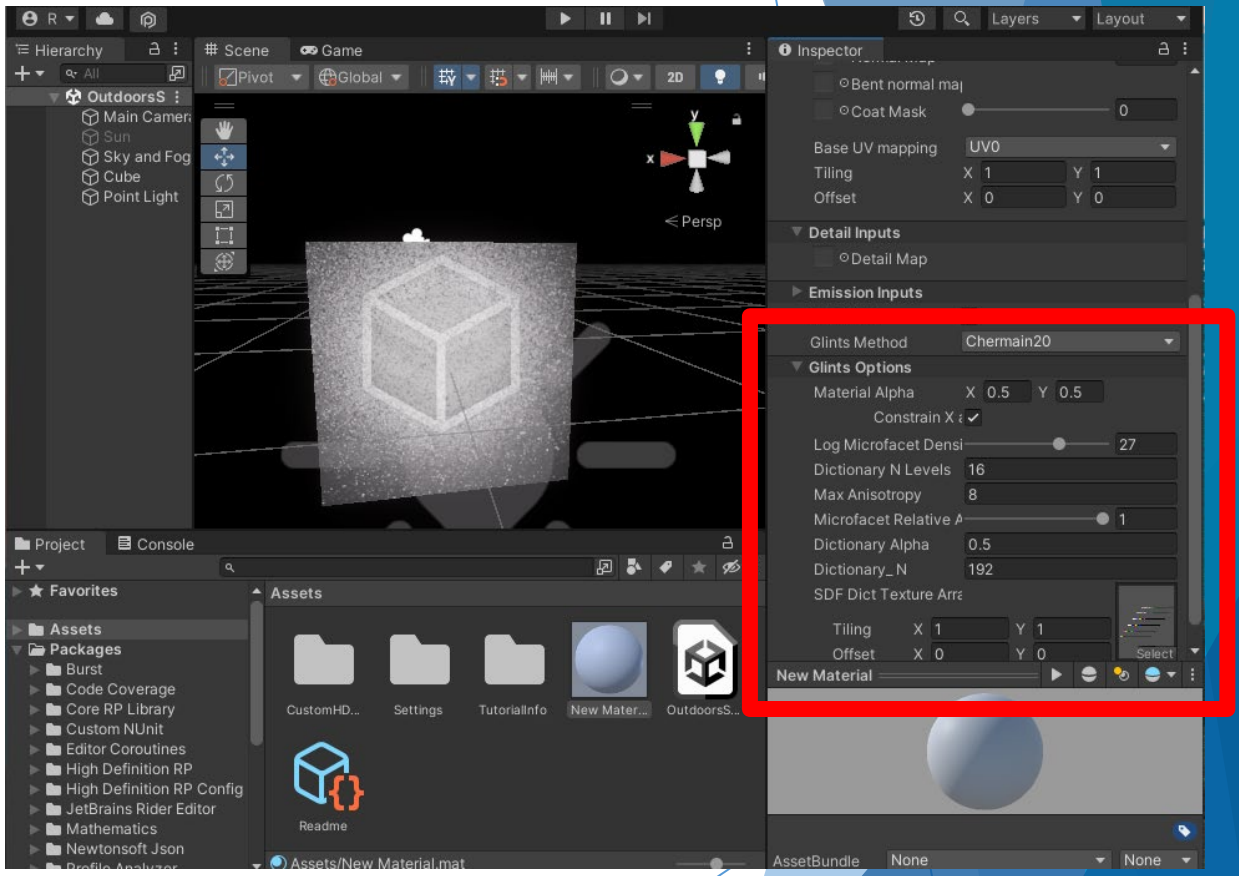


Tvorba upraveného Lit materiálu

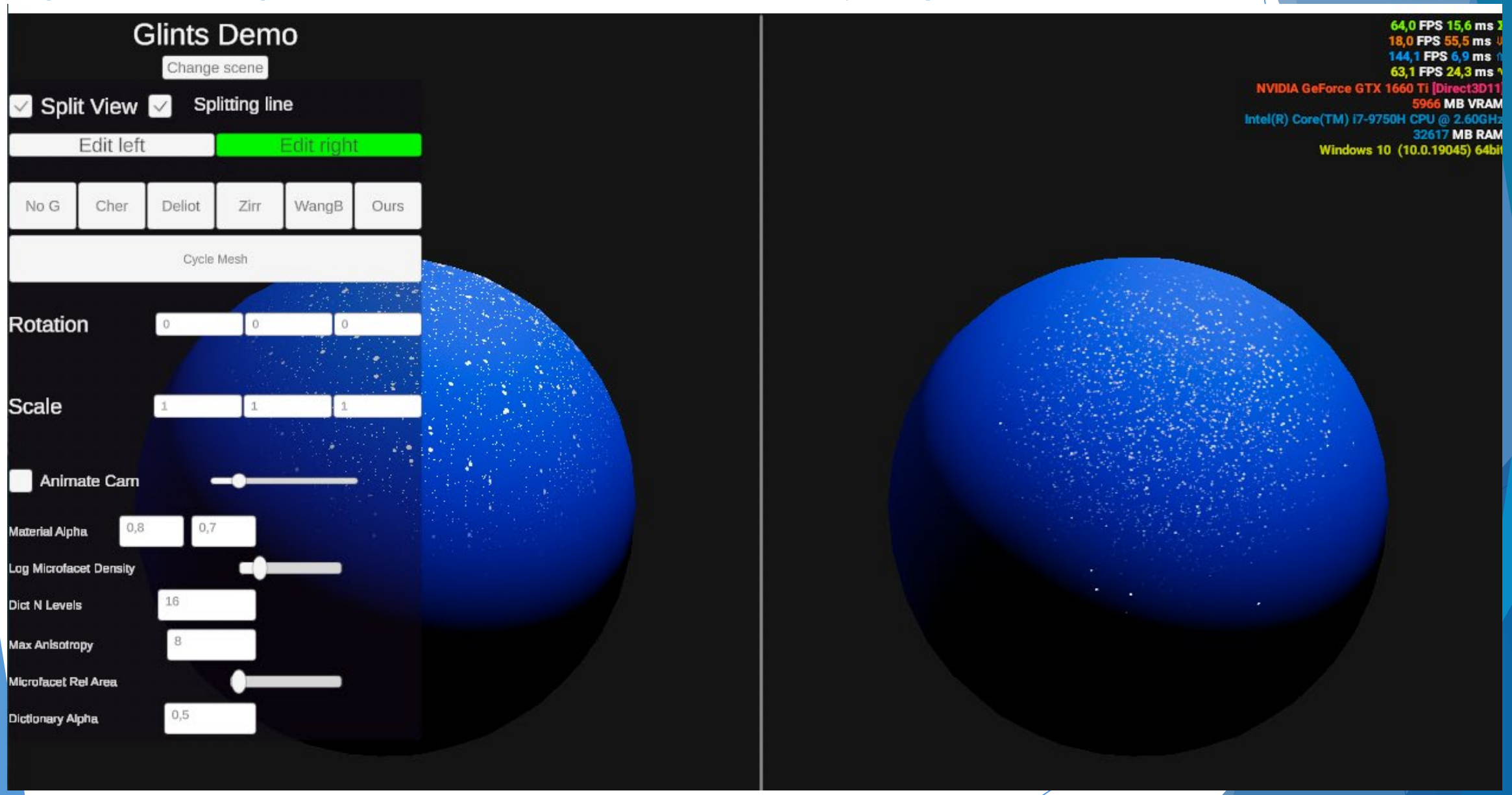
- Úprava HDRP vykresľovacieho kanála
  - Lokálna kópia Lit shaderu + modifikácie
- Možnosť vybrať 6 konfigurácií trblietok:
  - Bez + 4 existujúce metódy + naša vlastná metóda
- Sprístupnenie parametrov trblietok pre dané metódy cez vlastný editor
- Export ako samostatný stiahnuteľný shader v balíku
- Demo aplikácia na porovnanie všetkých metód s možnosťou zmeny parametrov

GUI materiálu, editor trblietok



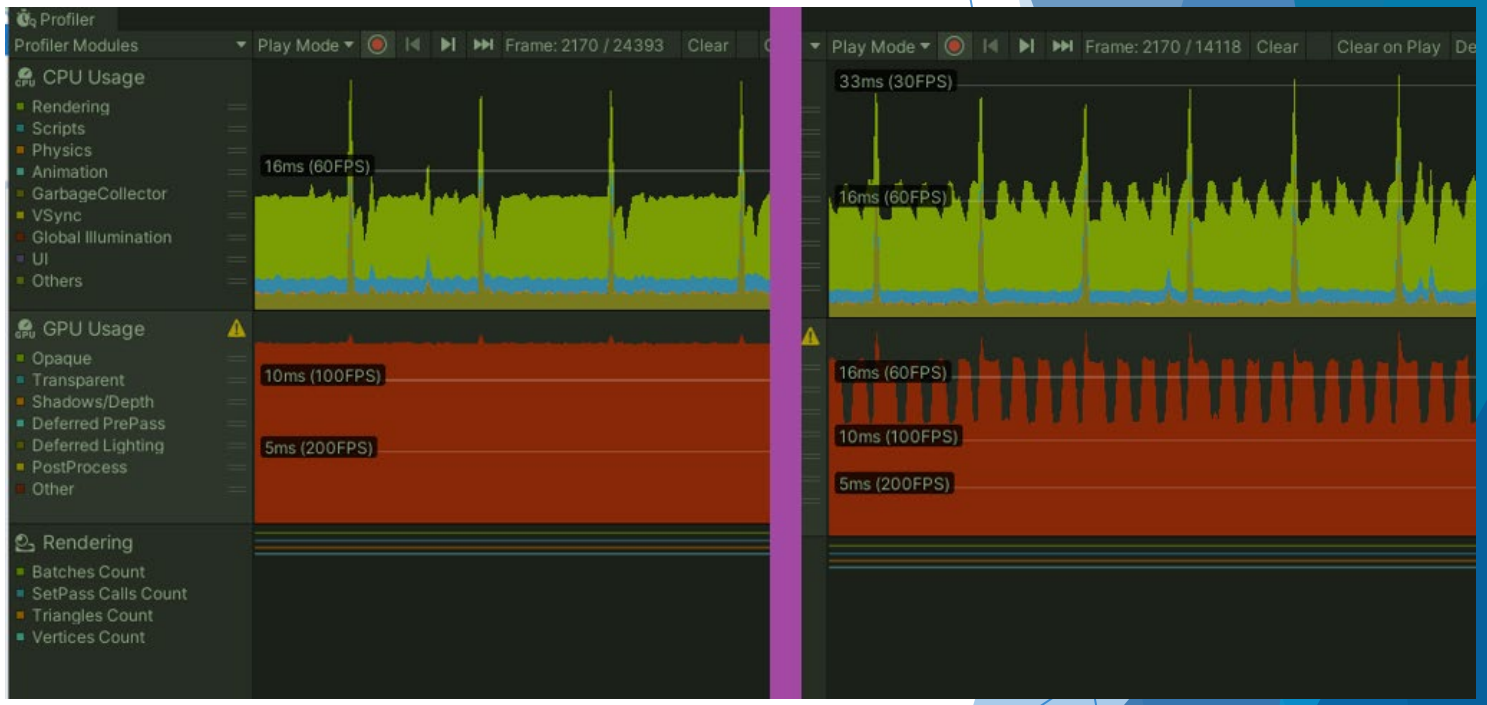


Aplikácia na porovnanie metód s nastaviteľnými parametrami



Obmedzenia vlastného materiálu

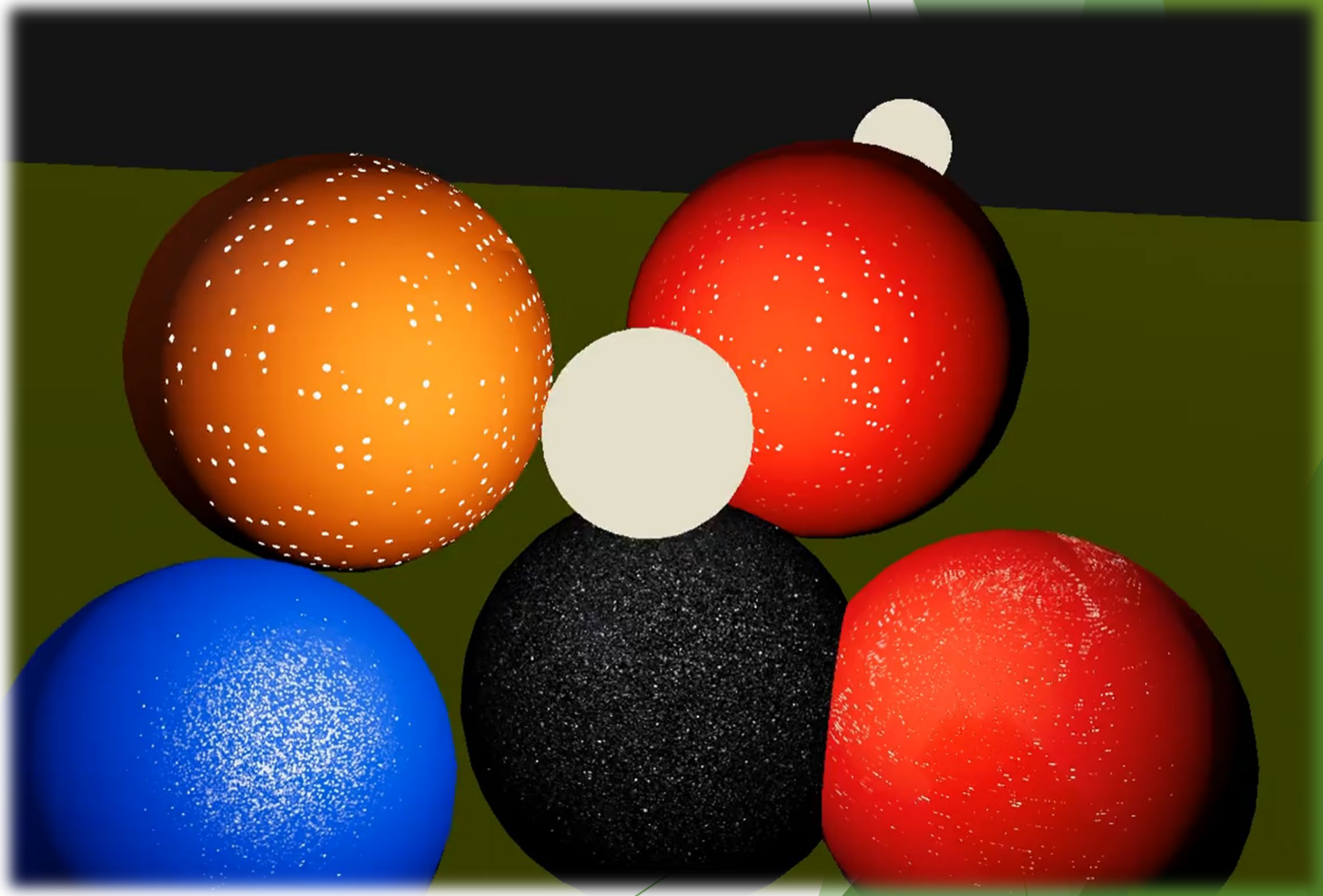
- Unity 2022.1 + HDRP 13.1.x
- Iba dopredné (forward) nasvietenie
- Iba bodové svetlo
- Nekonzistencia snímkovania
- Malá kontrola nad počtom svetiel, ktoré na materiál vplývajú



Unity profiler: Lit.shader (naľavo) je konzistentný, náš upravený CustomLit má špice

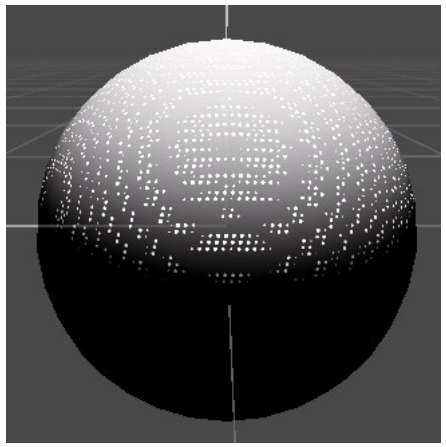
Implementované metódy trblietania

- 4 známe metódy:
  - Wang et al.(2015) (oranžová)
  - Zirr et al.(2016) (červená, dole)
  - Chermain et al.(2020) (modrá)
  - Deliot et al.(2023) (čierna)
- Naša metóda: rozšírenie Wang
  - červená, hore

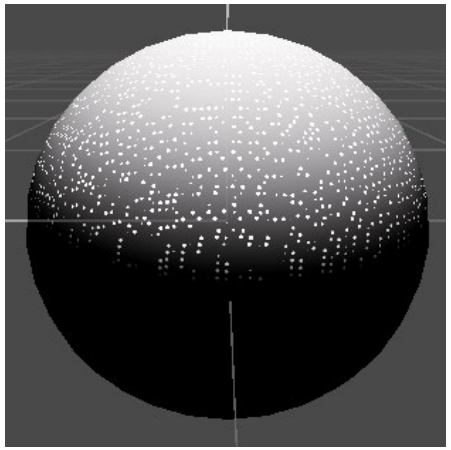


Vlastná metóda trblietania

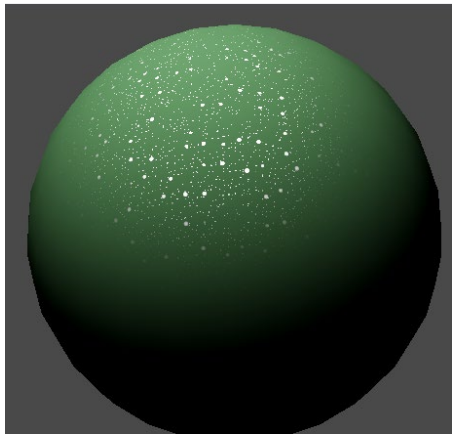
- Základ metódy Beibei Wang a Huv Bowles
- Hustejšie trblietky, lepšia variácia, škály trblietok
- Parameter globálna drsnosť
- Odstránené viditeľné vzory



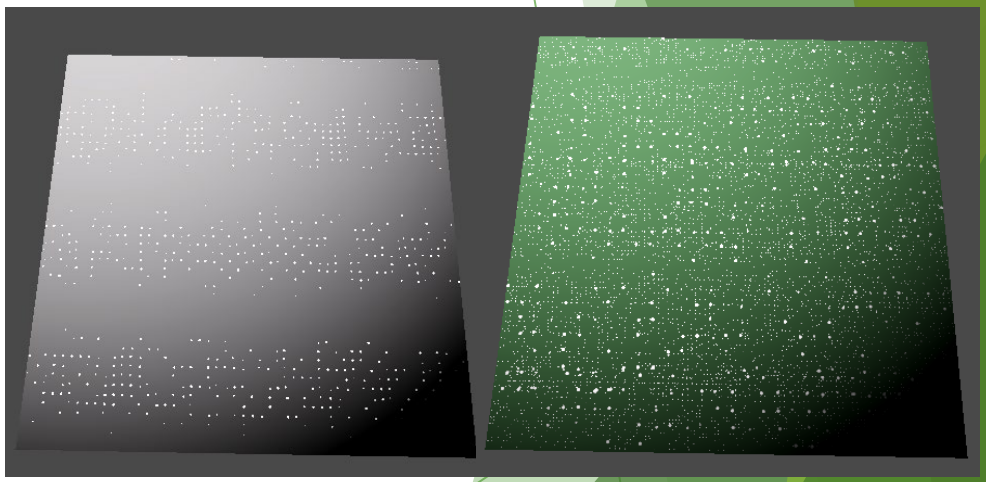
trblietky Wang bez šumu



Wang so šumom



naša metóda



trblietky Wang

naša metóda

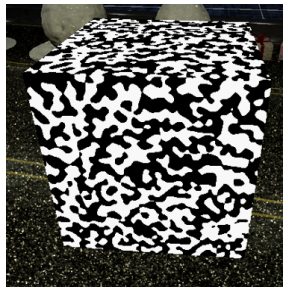
Vylepšenia

- Distribúcia trblietok:
  - hustejšie trblietky bez viditeľných vzorov
  - viaceré mriežky, náhodne posunuté
  - Škály trblietok- odstránenie skokov medzi úrovňami
- lepšia variácia
  - perturbácia mriežok, škál
- globálna drsnosť – Wardov anisotropný model

```
float3 GlintFade()
{
    float level = CalculateLevelBasedOnDistance();
    sparkleGridDensity *= level;
    noiseDensity *= level;

    float3 randomPosition = randomVec3(vObjPos.xy, inoise);
    float3 positionPlusView = objPos * sparkleGridDensity + wbeStruct.viewAmount *
        normalize(warpedViewDir + randomPosition);

    float3 gridCenter = GenerateGridCenter();
    float3 newGridOffset = positionPlusView - floor(positionPlusView);
    newGridOffset += JitterGrid();
    newGridOffset = Anisotropy();

    return float3(CalculateFinalContribution());
}
float3 CalcGrid(int i)
{
    float level0 = -1.0f;
    float level1 = 0.0f;
    float level2 = 1.0f;

    float3 resultC = GlintFade(level1);
    if(useScales)
        resultC += GlintFade(level0)+glintFade(level2);
}
float3 WBEnhancedGlints(float3 vObjPos, float3 vNormal, float3 lightPos, float3 vViewVec,
    float3 tangentVS, WBEStruct wbeStruct)
{
    float3 glittering = float3(0.0f, 0.0f, 0.0f);
    float specularity = WardAniso(ldir, vNormal, -n_view_dir, tangentVS);

        UNITY_LOOP for (int i = 1; 0 < _wbGridAmount; i++)
        {
            glittering += CalcGrid(i);
        }
    return glittering * specularity;
}
```

Obmedzenia vlastnej metódy

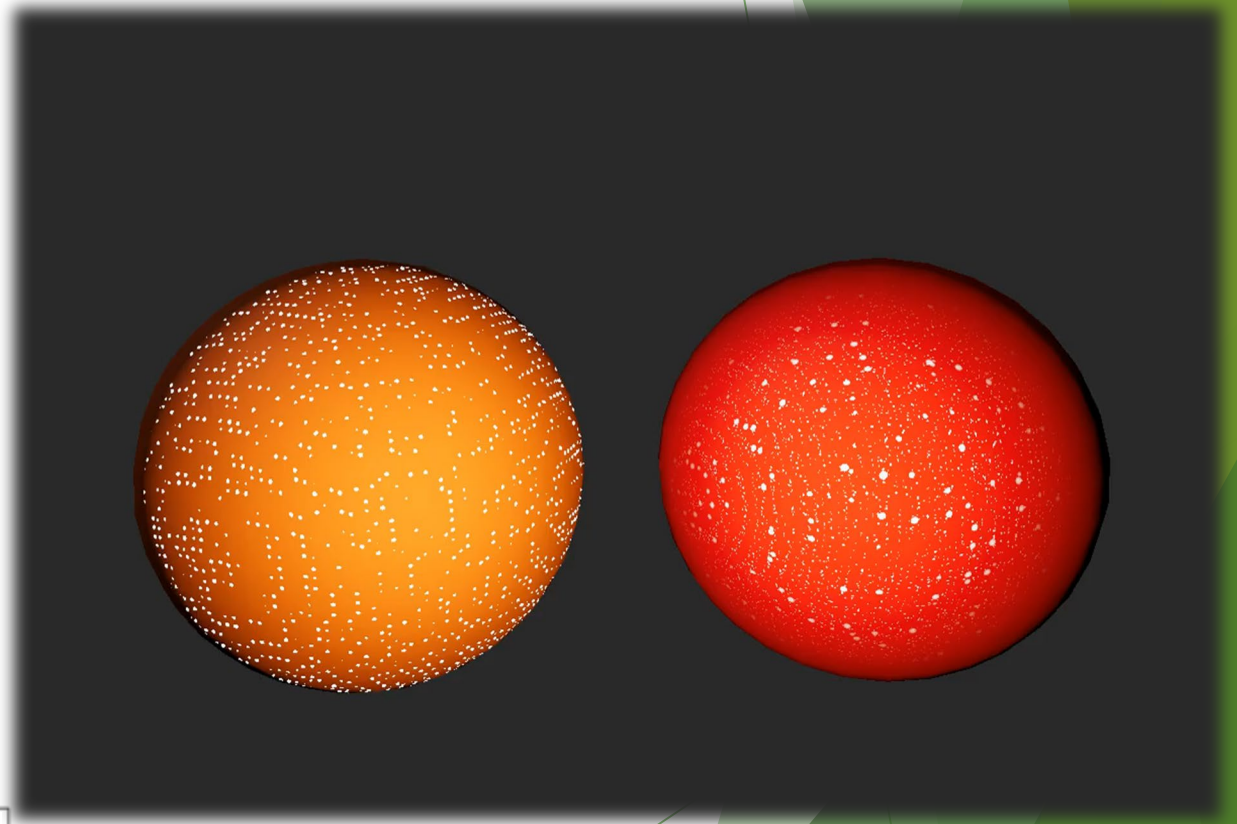
- Pri pohľade zblízka nepekné
- Príliš drobné trblietky sa mihotajú
- Viac mriežok = náročnejšie
- Viac svetiel = náročnejšie
- Stále sa dá všimnúť skoky
- Procedurálny šum je náročný:



Perlinov 3D šum

Metóda	priemer	max	min	99. percentil	99.9 percentil
Naša s procedurálnym šumom	44.8	65.8	32.4	67.0	76.6
Naša bez šumu	25.3	32.4	19.5	35.2	37.9

Tabuľka 4.5: Porovnanie času vykresľovania (v milisekundách) našej metódy pri použití šumu vs bez šumu.

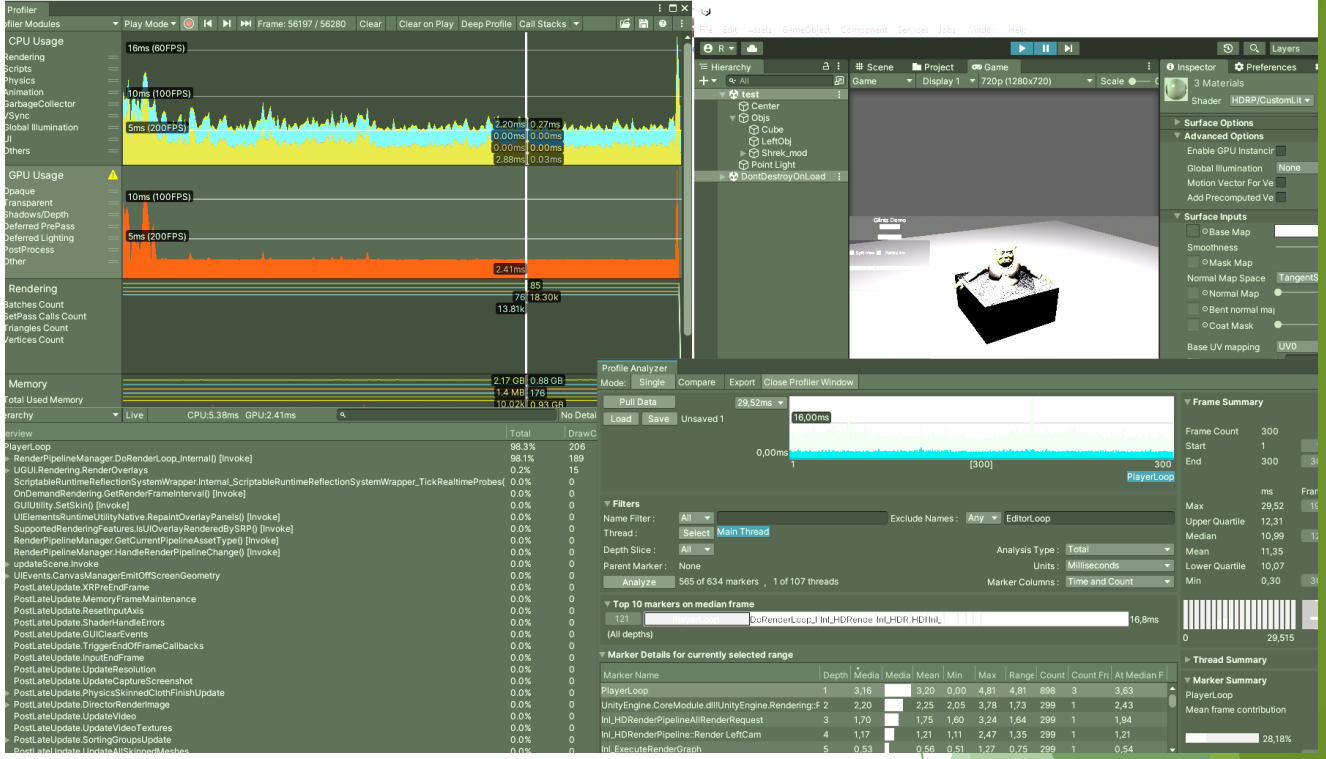


Metóda Wang

Naša metóda

Vyhodnotenie

- 3 testovacie scenáre – výkonnostná náročnosť
- Vyhodnotenie vizuálu
- Obmedzenia metód
- Jednoduchosť implementácie a použitia



Proces vyhodnocovania v Unity

Metóda	priemerné ms	max ms	min ms	99. percentil	99.9 percentil
Lit	17.2 ms	19.8 ms	14.1 ms	20.2 ms	20.7 ms
CustomLit Bez	17.5 ms	21.1 ms	13.0 ms	22.3 ms	22.8 ms
Chermain	47.2 ms	65.8 ms	28.6 ms	67.1 ms	67.2 ms
Deliot	48.1 ms	66.2 ms	32.5 ms	67.0 ms	67.0 ms
Zirr	42.0 ms	62.8 ms	26.7 ms	65.2 ms	66.5 ms
Wang	18.5 ms	23.7 ms	13.5 ms	23.7 ms	106.8 ms
Naša	44.8 ms	65.8 ms	32.4 ms	67.0 ms	76.6 ms

Tabuľka 4.3: Snímková frekvencia jednotlivých metód v ms pre scenár 3.

4K rozlíšenie,

16.6 ms ≈ 60 FPS

67 ms ≈ 14,9 FPS

Ďalšie zlepšenia

- podpora pre viac typov svetiel
- urýchlenie výpočtu perlinovho 3D šumu
- oprava nekonzistencie snímkovania
- pridať svetlu možnosť vylúčenia z vykreslenia trblietok(teraz môžu byť 3 vrstvy svetiel a čo ovplyv. - iba ostatné, iba trblietavé, aj-aj)

- skúsiť našu metódu spojiť s Deliot et al.- naša v diaľke(menej náročná)/ Deliot pri pohľade zblízka
- preskúmať metódu trblietania vhodnú pre sledovanie lúča / cesty v reálnom čase

Použité existujúce metódy

- *Wang Beibei a Bowles Huw. „A robust and flexible real-time sparkle effect“. V: EGSR 2016 E&I-Eurographics Symposium on Rendering-Experimental Ideas & Implementations. The Eurographics Association. 2016, s. 49–54.*
- *Zirr Tobias a Kaplanyan Anton S. „Real-time rendering of procedural multiscale materials“. V: Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. 2016, s. 139–148.*
- *Chermain Xavier, Sauvage Basile, Dischler J.-M. a Dachsbacher Carsten. „Procedural Physically based BRDF for Real-Time Rendering of Glints“. V: Computer Graphics Forum. Zv. 39. 7. Wiley Online Library. 2020, s. 243–253.*
- *Deliot Thomas a Belcour Laurent. „Real-Time Rendering of Glinty Appearances using Distributed Binomial Laws on Anisotropic Grids“. V: Computer Graphics Forum 42.8 (2023). issn: 1467-8659. doi: 10.1111/cgf.14866*

Ďakujem za Vašu pozornosť!