

Comparative analysis of the Cycles and Eevee graphics engines on the example of rendering 3D models of archaeological artifacts

Analiza porównawcza silników graficznych Cycles oraz Eevee na przykładzie renderowania modeli 3D artefaktów archeologicznych

Sebastian Dudek* , Krzysztof Dziedzic

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The purpose of the article is to conduct a comparative analysis of two graphics engines available in the Blender application. These are the Eevee and Cycles engines. The analysis will concern such elements as: efficiency, the ability to adjust the scene and the speed of image rendering by both engines. A test stand in the form of a desktop computer with a newly installed operating system has been prepared for the research.

Keywords: Blender; performance; graphic engine; comparison

Streszczenie

Celem artykułu jest przeprowadzenie analizy porównawczej dwóch silników graficznych dostępnych w aplikacji Blender. Są to silniki Eevee oraz Cycles. Analiza dotyczyć będzie takich elementów jak: wydajność, możliwość dostosowania sceny oraz szybkość renderowania obrazu przez oba silniki. Do przeprowadzenia badań przygotowane zostało stanowisko badawcze w postaci komputera stacjonarnego z nowo zainstalowanym systemem operacyjnym.

Słowa kluczowe: Blender; wydajność; silnik graficzny; porównanie

*Corresponding author

Email address: sebastian.dudek@pollub.edu.com (S. Dudek), k.diedzic@pollub.pl (K. Dziedzic)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Badania wydajności silników graficznych są obecne w literaturze naukowej. Jednymi z częściej porównywanych silników graficznych są Unreal Engine oraz Unity. Paweł Skop w swoim artykule z 2017 roku porównuje oba te silniki pod kątem wydajności na wybranych platformach. Na potrzeby artykułu dla obu silników stworzone zostały gry w wirtualnej rzeczywistości. Analiza wyników wykazała, że dostępne w momencie pracy nad artykułem ówczesne wersje tych popularnych silników różnią się od siebie ilością uzyskiwanych klatek obrazu na sekundę. Silnik Unity uzyskał lepsze rezultaty niż silnik Unreal Engine. Wyniki jakie uzyskano były odwrotne do tezy, w której autor zakładał, że to silnik Unity będzie charakteryzował się mniejszą wydajnością [1]. Przeprowadzone zostało również porównanie wydajności silnika Unity do Cry Engine, który również jest używany najczęściej w grach komputerowych [2]. Badania takie przeprowadził Huber Żukowski w 2019 roku. W swoim artykule porównywał on ilości uzyskanych klatek obrazu na sekundę, obciążenie procesora, oraz zajętość RAM. Również w tym przypadku silnik Unity okazał się rozwiązaniem wydajniejszym.

W erze, w której zdecydowana większość przemysłu filmowego wykorzystuje efekty komputerowe, projektowane gry stają się coraz bardziej rozbudowane i dopracowane, zwiększają się wymagania względem aplikacji wykorzystywanych do opracowania wszelkich materiałów graficznych [3]. Aby przenieść wizję artystów na ekrany potrzebne są odpowiednie narzędzia, które wspomagają i ułatwiają im pracę w szerokim

zakresie zastosowań grafiki komputerowej. Dostępne dziś narzędzia i rozwiązania dają szansę wejścia w świat grafiki komputerowej nie tylko największym firmom, ale też każdemu użytkownikowi, który korzysta z komputera w domu. Dostępne są bezpłatne rozwiązania, które dają nieograniczone możliwości twórcze. Zainteresowane osoby nie muszą więc inwestować dodatkowych środków, aby sprawdzić siebie w tej dziedzinie lub możliwości oprogramowania. Tak szeroki dostęp może zachęcić zainteresowanych do zrozumienia i poszerzenia swoich umiejętności w tej dziedzinie.

Możliwości graficzne w grach wideo z roku na rok wzrastają ze względu na rosnącą moc obliczeniową komputerów domowych. Twórcy już od jakiegoś czasu wdrażają w swoich silnikach, grach ray tracing. Ta technika pozwala zwiększyć realizm obrazu dzięki ulepszonym odbiciom na powierzchniach metalicznych lub połyskujących, cieniowaniu i okluzji otoczenia. Jeszcze do niedawna ray tracing był zarezerwowany głównie do tworzenia fotorealistycznych wizualizacji wnętrz, reklam lub w przemyśle filmowym. Wynikało to z braku wystarczającej mocy obliczeniowej, aby wygenerować odpowiednią liczbę klatek obrazu na sekundę do utrzymania płynności obrazu na pożądanym poziomie. Wykorzystując nowoczesny sprzęt, fotogrametrię lub skanery 3D możemy również skanować i przenosić obiekty niemal jeden do jednego do programów graficznych [4]. Obiekt taki możemy skanować w różnych rozdzielczościach, co zwiększa możliwość dostosowania go do założonych wymagań.

Historia grafiki komputerowej rozpoczyna się w dwudziestym wieku, a dokładnie są to lata sześćdziesiąte [5, 6]. Początkowo grafika komputerowa była zarezerwowana głównie dla wojska lub dużych firm z odpowiednio wysokim budżetem. Armia wykorzystywała tę dziedzinę do symulacji otoczenia w symulatorach lotów. Pojęcie grafiki komputerowej pojawiło się dzięki panu Williamowi Fetterowi. Był on artystą, który pracował w fabryce samolotów. Pierwszym narzędziem pozwalającym na pracę z bryłami oraz wykorzystanie linii był Sketchpad. Powstanie tej aplikacji można uznać za przełom w grafice komputerowej. Następne lata to czas rozwoju technologii i zwiększania się możliwości sprzętu. W latach siedemdziesiątych stworzony został pierwszy model ludzkiej twarzy. Był on również animowany. Jednym z pierwszych programów do modelowania w trzech wymiarach był 3D Core Graphic System. Grafika komputerowa coraz śmieiej wkraczała w świat reklamy, plakatów czy ulotek. Jedną z większych zmian było stworzenie przez firmę Apple komputera o nazwie Macintosh II. Był to komputer osobisty do użytku domowego. Posiadał on również rozbudowany program graficzny, którym był Adobe Photoshop. Aplikacja ta pozwalała na tworzenie grafik z wykorzystaniem warstw. Dzięki temu istniała i istnieje do dzisiaj możliwość niezależnej edycji poszczególnych sekcji tworzonego obrazu. W kolejnych latach grafika komputerowa stawała się coraz większą gałęzią komputerowego świata. Lata osiemdziesiąte i dziewięćdziesiąte to rozwój grafiki trójwymiarowej. W tym czasie powstaje aplikacja Blender. Pierwsza wersja, czyli Blender 1.0 została opublikowana w styczniu 1995 roku. Projekt jest rozwijany do dzisiaj. Obecna najnowsza wersja produkcyjna dostępna na stronie producenta to Blender 3.1.2.

2. Aplikacja Blender

Blender, widoczny na rysunku 1 jest pakietem narzędzi, który rozwijany jest obecnie przez Blender Foundation [7]. Oprogramowanie jest darmowe oraz udostępnione na licencji GNU GPL. Daje możliwość tworzenia animacji obiektów trójwymiarowych i dwuwymiarowych oraz renderingu przygotowanych scen. W aplikacji Blender zaimplementowano wiele przydatnych narzędzi. Pomagają i ułatwiają przeniesienie wizji artysty lub projektanta na ekran komputera [8]. Z ich pomocą obiekty trójwymiarowe mogą być poddane licznym modyfikacjom.



Rysunek 1: Interfejs aplikacji Blender 3.0.

Silnik Cycles od swojego debiutu na rynku przeszedł wiele zmian [9]. Po 10 latach istnienia wraz z Blenderem w wersji 3.0 pojawiła się nowa odsłona tego silnika o nazwie Cycles-X. Wprowadzonych zostało wiele zmian, które na celu miały zastąpienie starych i nieoptymalizowanych rozwiązań oraz algorytmów tego silnika graficznego. Pozwoliło to znacząco zwiększyć wydajność Cycles. Zmiany przyniosły nawet o 50% szybsze czasy renderowania w stosunku do możliwości jakie oferowała poprzednia wersja silnika. Dodana została obsługa interfejsu API NVIDIA OptiX. Kiedy na rynku pojawiły się karty graficzne z rodziny GeForce Turing, pokazano też pierwsze modele z dedykowanymi jednostkami RT. Jednostki te mogą wspomagać i skracać czas w obliczeniach związanych ze śledzeniem promieni. Dającym duże możliwości poprawy obrazu jest wbudowany w aplikację Blender denoiser. Jest to narzędzie służące do odszumiania obrazu. Podczas renderowania, najczęściej w ciemnych miejscach powstaje szum. Może być to spowodowane niskim próbkowaniem w ustawieniach naszego silnika renderującego. Denoiser ma na celu usunięcie powstałego szumu za pomocą technik przetwarzania końcowego, które oparte są na uczeniu maszynowym. Ma on jednak swoje wady. Jeżeli próbkowanie będzie za niskie, a informacji na obrazie zbyt mało, efekt końcowy będzie lekko niewyraźny, rozmazany.

Silnik Eevee został wprowadzony w jednej z aktualizacji do aplikacji Blender, a dokładnie do wersji 2.80 [10]. Przy okazji tej zmiany odświeżony został również interfejs omawianego oprogramowania. Silnik ten działa na podobnej zasadzie co inne silniki renderujące w czasie rzeczywistym. Rozwiązania takie świetnie sprawdzają się w grach komputerowych. Przykładami konkurencyjnych silników mogą być Unreal Engine oraz Unity. Technologia i sposób generowania obrazu sprawia, że silnik ten w porównaniu do Cycles będzie generował gorszej jakości odbicia, cieniowanie czy efekty wolumetryczne. Eevee nie wykorzystuje ray tracingu. Jego przewagą jest znacznie, krótszy czas renderowania obrazów. Dużą zaletą jest też to, że obraz w podglądzie jest identyczny jak końcowy rezultat po wyrenderowaniu. Daje to możliwość wprowadzania poprawek w modelach, teksturach czy oświetleniu na bieżąco bez konieczności częstego renderowania sceny, aby zobaczyć efekt końcowy. Dzięki temu zaoszczędzony czas można poświęcić na inne aspekty projektu. Silnik o takiej specyfice jest przydatny i wygodniejszy w momencie tworzenia animacji. Po raz kolejny oferowana szybkość pozwala na lepszą kontrolę nad animacjami. Nie jest konieczne renderowanie czasami wielu setek czy nawet tysięcy klatek, aby móc wychwycić niedoskonałości. Eevee wspiera technikę Physically Based Rendering (PBR).

3. Opis badania

3.1. Obiekty badań

Do przeprowadzenia badań zeskanowane zostały rzeczywiste obiekty, artefakty archeologiczne. Przygotowane zostały różne ich wersje o różnej ilości trójkątów

oraz rozdzielczości tekstur. Za pomocą skanera na światło strukturalne uzyskana została chmura punktów, która wykorzystana została do przygotowania modeli 3D. użytym skanerem był Artec EVA (Rysunek 2)[11].



Rysunek 2: Skaner Artec EVA.

Testy wykonano na trzech obiektach archeologicznych. Każdy z nich posiadał trzy warianty o różnym zagęszczeniu siatki:

- 500 wierzchołków,
- 3000 wierzchołków,
- 6000 wierzchołków.



Rysunek 3: Zeskanowany model, Model_01, 6000 wierzchołków.

Dla zeskanowanych obiektów przygotowane zostały trzy warianty tekstur o różnej rozdzielczości:

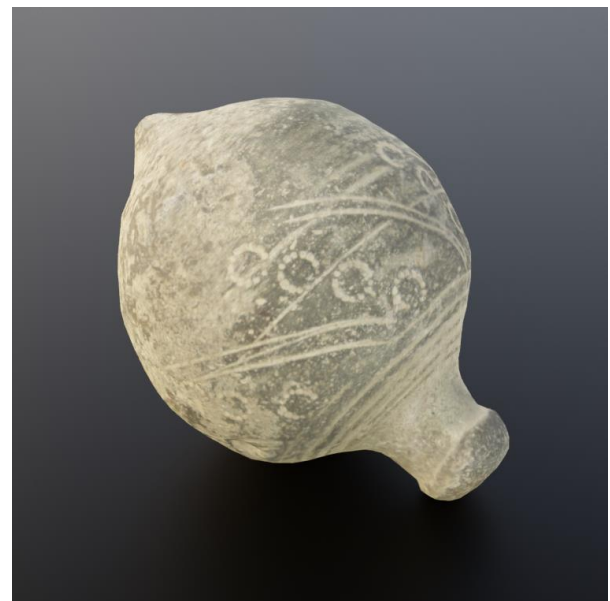
- 512 x 512 pikseli,
- 1024 x 1024 pikseli,
- 2048 x 2048 pikseli.

Zeskanowane modele przedstawione na Rysunkach 3, 4 oraz 5 zostały umieszczone na prostej w budowie scenie. Aby wykluczyć wpływ obiektów otaczających

badane warianty modeli, środowisko jest maksymalnie uproszczone. Gotowa scena składa się z zeskanowanego obiektu w wybranym wariantcie, podłogi w postaci obiektu plane, dostępnego domyślnie w aplikacji Blender oraz mapy HDRI o rozdzielczości 2160p.



Rysunek 4: Zeskanowany model, Model_02, 500 wierzchołków.



Rysunek 5: Zeskanowany model, Model_03, 3000 wierzchołków.

3.2. Metoda badań

Badanie wydajności polegało na sprawdzeniu czasów renderowania, danej sceny wraz z obiektami o różnej ilości wierzchołków. Dla każdego modelu o danej jakości wykonanych zostało 5 prób. Uzyskane czasy zapisywane były na podstawie czasomierza wbudowanego w aplikację. Każda próba renderowania była wykonana w ten sam sposób. Mając obiekt i scenę w podglądzie aplikacji Blender, używany był klawisz F12 odpowiedzialny za rozpoczęcie procesu renderowania. Po zakończonej operacji powracano ponownie do widoku podglądu i renderowanie rozpoczynano od nowa w

analogiczny sposób. Badanie różnic w generowanym obrazie polegało na porównywaniu dwóch wyrenderowanych przy użyciu silników Cycles oraz Eevee scen. Analizie podlegało cieniowanie, odbicia oraz oświetlenie.

3.3. Stanowisko badawcze

Badania przeprowadzone zostały na komputerze o specyfikacji technicznej podanej w tabeli (Tabela 1).

Tabela 1: Specyfikacja techniczna stanowiska badawczego

Podzespół	Model/Parametry
Procesor	AMD Ryzen 3600 6/12, 4,2GHz
Karta graficzna	Palit RTX 3060TI 8GB
Dysk SSD 1	Adata 8200SX Pro
Dysk SSD 2	Samsung SSD 870 QVO 1TB
Płyta główna	Asus ROG Strix B450-F Gaming
RAM	16GB (2x8GB) DDR4 3200 MHz
System operacyjny	MS Windows 11 Pro 64bit

4. Wyniki badań

Wyniki testów przeprowadzonych zgodnie ze scenariuszem badawczym przedstawiają tabele. Tabele z wynikami przedstawiają czasy renderowania dla silnika Cycles (Tabela 2) oraz Eevee (Tabela 3).

Tabela 2: Czasy renderowania dla silnika Cycles.

Model_01		
500 verts	3000 verts	6000 verts
13,25[s]	13,49[s]	13,64[s]
13,33[s]	13,30[s]	13,50[s]
13,25[s]	13,42[s]	13,49[s]
13,18[s]	13,40[s]	13,42[s]
13,13[s]	13,31[s]	13,46[s]
Model_02		
500 verts	3000 verts	6000 verts
13,27[s]	13,40[s]	13,56[s]
13,04[s]	13,43[s]	13,32[s]
12,86[s]	13,28[s]	13,45[s]
13,02[s]	13,23[s]	13,41[s]
13,09[s]	13,21[s]	13,33[s]
Model_03		
500 verts	3000 verts	6000 verts
12,30[s]	12,38[s]	12,40[s]
12,18[s]	12,29[s]	12,43[s]
12,10[s]	12,21[s]	12,25[s]
12,06[s]	12,09[s]	12,45[s]
12,08[s]	12,26[s]	12,32[s]

Tabela 3: Czasy renderowania dla silnika Eevee.

Model_01		
500 verts	3000 verts	6000 verts
0,76[s]	0,90[s]	0,97[s]
0,80[s]	0,81[s]	0,72[s]
0,88[s]	0,78[s]	0,81[s]
0,75[s]	0,75[s]	0,84[s]
0,71[s]	0,84[s]	0,79[s]
Model_02		

500 verts	3000 verts	6000 verts
0,81[s]	0,89[s]	0,91[s]
0,79[s]	0,71[s]	0,77[s]
0,78[s]	0,79[s]	0,81[s]
0,71[s]	0,73[s]	0,74[s]
0,79[s]	0,81[s]	0,82[s]
Model_03		
500 verts	3000 verts	6000 verts
0,65[s]	0,75[s]	0,77[s]
0,70[s]	0,84[s]	0,86[s]
0,69[s]	0,71[s]	0,81[s]
0,76[s]	0,83[s]	0,79[s]
0,75[s]	0,79[s]	0,85[s]

Po przeprowadzeniu badań wydajności, stworzona została dodatkowa scena pozwalająca na porównanie obu silników pod względem jakości generowanego obrazu. Na Rysunku 6 przedstawiono wykorzystane do badań trzy modele obiektów archeologicznych wkomponowane w wymodelowaną scenę. Nowa scena składała się z biblioteczki, podłoża, w którego skład wchodziła roślinność oraz kamienie, dodatkowo wymodelowane zostały elementy dekoracyjne otaczające zeskanowane obiekty archeologiczne. Za tło posłużył obiekt o nazwie plane, prostokąt złożony z czterech wierzchołków, z nałożoną teksturą gór. Użyta została również mapa HDRI generująca ostrzejsze światło i cienie. Rysunek podzielony został na cztery części. Widoczny jest obraz wygenerowany przez silnik Eevee oraz efekt pracy silnika Cycles.



Rysunek 6: Różnice w generowaniu obrazu przez badane silniki.

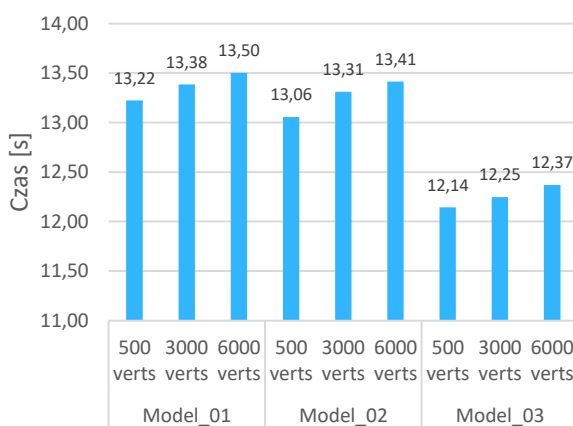
W tabeli (Tabela 4) przedstawione zostały czasy renderowania sceny dla różnych wariantów tekstur. Badanie zostało przeprowadzone na silniku Cycles. Scena składała się z dwóch obiektów. Użyty został Model_01, który składał się z 6000 wierzchołów, oraz tło w postaci obiektu plane dostępnego natywnie w aplikacji Blender.

Tabela 4: Pomiar czasu renderowania dla różnych wariantów tekstur

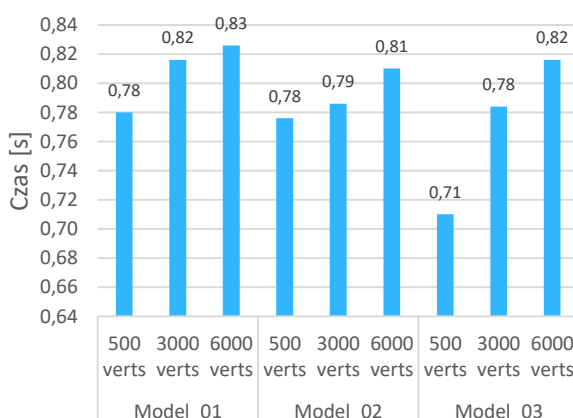
Model_01, 6000 verts		
512 x 512	1024 x 1024	2048 x 2048
13,66[s]	13,74[s]	13,66[s]
13,64[s]	13,59[s]	13,69[s]
13,63[s]	13,67[s]	13,58[s]
13,73[s]	13,65[s]	13,67[s]
13,62[s]	13,60[s]	13,69[s]

5. Analiza wyników

Uzyskane wyniki pozwoliły na obliczenie średnich czasów renderowania dla każdego przypadku. Na Rysunku 7 i 8 przedstawiono średnie wyniki obu silników graficznych. Do wyliczenia średniej użyto 5 wyników na dany wariant modelu.



Rysunek 7: Średni czas renderowania scen dla silnika Cycles.



Rysunek 8: Średni czas renderowania scen dla silnika Eevee.

Analizując wykresy oraz uzyskane wyniki stwierdzić można, że czas renderowania na silniku Cycles w stosunku do silnika Eevee jest znacznie dłuższy. Cycles uzyskał średni czas renderowania wszystkich scen na poziomie 12,96 s. Średni czas renderowania dla silnika

Eevee wynosił 0,79 s. Wynika to z wykonywania o wiele bardziej skomplikowanych obliczeń światła, okluzji otoczenia oraz cieni. Silnik Eevee ma za to bardzo dużą przewagę jeżeli głównym kryterium wyboru jest czas wyrenderowania sceny. Z obliczeń średnich czasów wynika, że Eevee pozwolił na skrócenie czasu renderowania średnio o 93%. Zwiększenie gęstości siatki miało niewielki wpływ na czas renderowania sceny w przypadku porównywanych silników i stworzonej sceny. Uzyskane wyniki świadczą o tym, że w przypadku silników różnica w ilości poligonów na scenie musiałaby być wielokrotnie wyższa, żeby można było zobaczyć znaczny wzrost czasu renderowania. Pomimo niewielkich wzrostów uzyskanych wyników, wykresy jednoznacznie pokazują, że czas oczekiwania na gotowy obraz zwiększa się wraz ze wzrostem poligonów badanych obiektów archeologicznych. Konieczność dodatkowych obliczeń dla zwiększonej ilości ścian obiektu wydłuża oczekiwanie na końcowy rezultat. Wykonywane są obliczenia dotyczące światła, cieni oraz obić, a wraz z zagęszczeniem siatki wzrasta ilość płaszczyzn dla których konieczne jest wykonanie tych obliczeń. Porównując dwa silniki graficzne Eevee oraz Cycles, widać różnicę między czasem wyrenderowania tej samej sceny na korzyść silnika Eevee. Ma to związek ze sposobem działania obu silników. Silnik Cycles wykorzystuje do renderowania obrazu technikę zwaną ray tracing. Operacje jakie wykonuje są dużo bardziej skomplikowane. Obliczenia dotyczą światła, odbić oraz cieniowania. Pozwala to uzyskać obraz z poprawnymi fizycznie opisanymi aspektami względem silnika Eevee, ale kosztem dużo dłuższego czasu renderowania.

Analiza uzyskanych wyników dla różnych wariantów tekstur wykazała, że w przyjętym scenariuszu zwiększenie ich rozdzielczości nie spowodowało wydłużenia czasu renderowania sceny. Różnica między najdłuższym (13,74[s]), a najkrótszym (13,58[s]) uzyskanym czasem dla trzech przygotowanych wariantów to 1,16%.

Analizę jakości obrazów warto rozpocząć od cieniowania. Na rysunku dostrzec można różnice w tym aspekcie, szczególnie we wnętrzu szafki. Silnik Eevee nie generuje cieni tak dokładnych jak Cycles lub brakuje ich w ogóle. Ustawione zostało oświetlenie na takie, które generuje ostrzejsze bardziej widoczne cienie [12]. Różnice dostrzec też można w cieniowaniu gdzie obiekty stykają się z podłożem. Silnik Eevee nie wygenerował cieni pod modelem szafki przez co obraz sprawia wrażenie płaskiego, bez głębi. Drugą różnicą jest oświetlenie i odbicie. Silnik Cycles na elementach połyskujących generował poprawne fizyczne odbicia czego nie robił silnik Eevee.

6. Podsumowanie

W artykule przedstawiono porównanie dwóch silników graficznych, Cycles oraz Eevee. Metoda badań wykazała, że czas renderowania przy użyciu silnika Cycles jest znacząco dłuższy niż na drugim badanym silniku. Oba silniki są również powtarzalne w uzyskiwanych wynikach. Przeprowadzone badania wykazały, że mając na

scenie określoną ilość wierzchołków, trójkątów możemy spodziewać się zbliżonych czasów renderowania. Wpływ zwiększenia jakości tekstur w obranym przypadku był nieznaczny.

Po bezpośrednim porównaniu i przeanalizowaniu tej samej sceny wyrenderowanej przez oba silniki stwierdzić można, że silnik Cycles generuje obraz wyższej jakości niż silnik Eevee. Jest to związane ze sposobem działania obu silników. Użyty w silniku Cycles ray tracing pozwala na uzyskanie poprawnych fizycznie odbić, cieni oraz efektów wolumetrycznych. Zaletą silnika Eevee w tym wypadku jest znacznie skrócony czas renderowania przygotowanych scen.

Literatura

- [1] P. Skop, Porównanie wydajności silników gier na wybranych platformach, *Journal of Computer Sciences Institute* 7 (2018) 116–119.
- [2] H. Żukowski, Porównanie wydajności trójwymiarowych gier z użyciem silników CryEngine i Unity, *Journal of Computer Sciences Institute*, 13 (2019) 345–348 <https://doi.org/10.35784/jcsi.1330>.
- [3] A. Thorn, *Unity i Blender. Praktyczne tworzenie gier*, Helion, Gliwice, 2015.
- [4] E. Głowienka, B. Jankowicz, B. Kwoczyńska, P. Kuras, K. Michałowska, S. Mikrut, A. Moskal, I. Piech, M. Strach, J. Sroka, *Fotogrametria i skaning laserowy w modelowaniu 3D*, pod redakcją Sławomira Mikruta Ewy Głowienki, Wyższa Szkoła Inżyniersko-Ekonomiczna z siedzibą w Rzeszowie (2015) 78–79.
- [5] P. Chlipalski, *Blender. Architektura i projektowanie*. Wydanie II, Helion, Gliwice, 2018.
- [6] D. Holcer, *Ewolucja grafiki komputerowej w świecie rozrywki interaktywnej*, Repozytorium Uniwersytetu Jagiellońskiego, praca magisterska, 2015, <https://ruj.uj.edu.pl/xmlui/handle/item/207622>.
- [7] B. Simonds, *Blender. Praktyczny przewodnik po modelowaniu, rzeźbieniu i renderowaniu*, Helion, Gliwice, 2014.
- [8] K. Kukło, J. Kolmaga, *Blender. Kompendium*, Helion, Gliwice, 2007.
- [9] Strona z przewodnikiem silnika Cycles: <https://docs.blender.org/manual/en/latest/render/cycles/index.html>, [06.04.2022].
- [10] Strona z przewodnikiem silnika Eevee: <https://docs.blender.org/manual/en/latest/render/eevee/index.html>, [06.04.2022].
- [11] Instrukcja obsługi skanera Artec Eva: <https://www.manualslib.com/manual/1755978/Artec-3d-Eva.html>, [04.04.2022].
- [12] J. Birn, *Cyfrowe oświetlenie i rendering*. Wydanie II, Helion, Gliwice, 2007.