

Jakub Sawa¹, dr inż. Marcin Klimek²

Akademia Bialska im. Jana Pawła II, Biała Podlaska

https://doi.org/10.29316/9788368103281_7

OCENA WYDAJNOŚCI PRZEGLĄDARKI OBIEKTÓW PRZESTRZENNYCH NA WYBRANYCH URZĄDZENIACH RZECZYWISTOŚCI MIESZANEJ I MOBILNYCH

EVALUATION OF SPATIAL OBJECT BROWSER PERFORMANCE ON SELECTED MIXED REALITY AND MOBILE DEVICES

Streszczenie

Rozdział przedstawia analizę wydajności przeglądarki obiektów przestrzennych uruchamianej na urządzeniach mobilnych oraz platformie rzeczywistości mieszanej. Badania przeprowadzono z wykorzystaniem autorskiej aplikacji testowej stworzonej w silniku Unity 2022.3 LTS. Celem analizy było określenie efektywności renderowania modeli trójwymiarowych o zróżnicowanej złożoności pod kątem utrzymania płynności animacji, obciążenia systemowego i stabilności termicznej urządzeń. Testy objęły urządzenia Meta Quest 3, Samsung Galaxy S20 FE, Samsung Galaxy Z Flip 3 oraz Xiaomi POCO F6. Zastosowano zestaw dziewięciu modeli 3D obejmujących zakres od 12 do 1 349 052 trójkątów, co umożliwiło identyfikację progów wydajnościowych dla poszczególnych

¹ student, e-mail: sawa23891@student.akademiabialska.pl

² e-mail: m.klimek@dyd.akademiabialska.pl

platform. Wyniki badań wskazują na różnice pomiędzy urządzeniami – najwydajniejszym okazał się POCO F6, który utrzymywał średnio 30.0 FPS obsługując wszystkie modele testowe. Meta Quest 3 wykazał systemowe ograniczenie do 24.0 FPS. Zidentyfikowano progi krytyczne wydajności przy 277 644 trójkątów dla urządzeń Samsung oraz przy 869 942 trójkątów dla POCO F6.

Słowa kluczowe: przeglądarka obiektów przestrzennych, renderowanie, urządzenia mobilne, rzeczywistość mieszana, Unity

Abstract

This chapter presents an analysis of the performance of a spatial object browser running on mobile devices and a mixed reality platform. The research was conducted using a proprietary test application created in Unity 2022.3 LTS. The aim of the analysis was to determine the rendering efficiency of three-dimensional models of varying complexity in terms of maintaining animation fluidity, system load and thermal stability of devices. The tests covered Meta Quest 3, Samsung Galaxy S20 FE, Samsung Galaxy Z Flip 3 and Xiaomi POCO F6 devices. A set of nine 3D models ranging from 12 to 1,349,052 triangles were used, which made it possible to identify performance thresholds for individual platforms. The results of the tests show differences between the devices – the POCO F6 proved to be the most efficient, maintaining an average of 30.0 FPS while running all test models. Meta Quest 3 showed a system limitation of 24.0 FPS. Critical performance thresholds were identified at 277,644 triangles for Samsung devices and 869,942 triangles for the POCO F6.

Keywords: spatial object browser, rendering, mobile devices, mixed reality, Unity

Wstęp

Technologie rzeczywistości rozszerzonej i mieszanej nabrały znaczenia w ciągu ostatnich lat dzięki rosnącej dostępności urządzeń o wysokiej wydajności obliczeniowej. Integracja technik renderowania 3D w urządzeniach mobilnych oraz autonomicznych headsetach umożliwiła tworzenie aplikacji, które dawniej wymagały komputerów stacjonarnych. Zwiększona moc procesorów i układów graficznych, m.in. serii Qualcomm Snapdragon, pozwala na płynne wizualizacje modeli o wysokiej szczegółowości.

Przeglądarki obiektów przestrzennych stanowią kluczowe narzędzie w wielu sektorach, w tym medycynie, architekturze, edukacji i przemyśle. Ich funkcje obejmują m.in. ładowanie modeli 3D, manipulację obiektami, analizę konstrukcji czy obsługę wizualizacji interaktywnych. W miarę wzrostu

oczekiwań użytkowników i wzrostu złożoności danych przestrzennych, coraz większą rolę odgrywa wydajność systemu oraz responsywność interfejsu.

Analiza dostępnej literatury wskazuje, że istniejące opracowania koncentrują się na pojedynczych platformach sprzętowych lub wykorzystują ograniczone zestawy modeli testowych. Analizowana jest wydajność aplikacji mobilnych (Nowak, Wiśniewski, 2021), ale bez uwzględnienia platform rzeczywistości mieszanej. Deweloperzy aplikacji XR potrzebują obiektywnych danych o wydajności różnych platform sprzętowych przy renderowaniu modeli o zróżnicowanej złożoności geometrycznej.

Kluczowym wyzwaniem metodologicznym stanowi porównanie wydajności między różnymi architekturami aplikacji. Pierwotnie planowana implementacja dual architecture natywne aplikacje VR dla urządzeń MR oraz osobne aplikacje mobilne dla smartfonów – wprowadza nieścisłości pomiarowe wynikające z różnic implementacyjnych. W dokumentacji Unity Technologies (Unity Technologies, 2022) znajduje się stwierdzenie, że porównywanie wyników wydajnościowych między różnymi architekturami aplikacji prowadzi do zniekształcenia wniosków badawczych.

Rozwiązaniem metodologicznym jest opracowanie jednolitej aplikacji testowej działającej w trybie 2D na systemie Android dla wszystkich platform docelowych. Takie podejście eliminuje zmienne związane z różnicami architektonicznymi i umożliwia bezpośrednie porównanie wpływu możliwości sprzętowych urządzeń na wydajność identycznej aplikacji. Meta Quest 3 testowane w trybie kompatybilności Android zachowuje porównywalność z urządzeniami mobilnymi przy wykorzystaniu zaawansowanych możliwości sprzętowych dedykowanego procesora Snapdragon XR2 Gen 2.

Celem pracy jest opracowanie aplikacji testowej do oceny wydajności przeglądarki obiektów przestrzennych oraz przeprowadzenie analizy porównawczej na wybranych urządzeniach mobilnych i rzeczywistości mieszanej. W ramach prac zaimplementowano system monitorowania parametrów wydajnościowych, przygotowano zestaw modeli testowych o zróżnicowanej złożoności geometrycznej oraz przeprowadzono pomiary na urządzeniach reprezentujących różne kategorie sprzętowe na urządzeniach z systemem Android: Xiaomi POCO F6 (Snapdragon 8s Gen 3, Adreno 735), Samsung Galaxy Z Flip 3 (Snapdragon 888, Adreno 660), Samsung Galaxy S20 FE

(Snapdragon 865, Adreno 650) oraz Meta Quest 3 (Snapdragon XR2 Gen 2, Adreno 740) w trybie kompatybilności Android.

Aby dokonać obiektywnej oceny wydajności przeglądarki obiektów przestrzennych na różnych platformach sprzętowych stworzono jedno środowisko operacyjne eliminując różnice wynikające ze specyfiki interfejsów XR. Jako metryki użyto: liczby klatek na sekundę FPS (ang. Frames Per Second, statystyka pozwalająca określić, ile nowych klatek jest wyświetlane w ciągu jednej sekundy), wykorzystanie CPU i GPU, zużycie RAM oraz temperatury urządzeń podczas renderowania modeli o zróżnicowanej złożoności geometrycznej.

Przegląd badań

Współczesne przeglądarki modeli 3D umożliwiają interaktywną prezentację obiektów, ich manipulację i analizę w czasie rzeczywistym. Są stosowane w wielu dziedzinach przemysłu, szczególnie tam, gdzie kluczowe jest przedstawienie struktury przestrzennej obiektu. Rozwój technologii mobilnych i XR umożliwił wykorzystanie przeglądarek poza tradycyjnymi systemami desktopowymi.

Przeglądarki te wymagają bardzo efektywnego zarządzania zasobami, ponieważ operują na modelach o złożonej geometrii, często wyposażonych w szczegółowe tekstury i zaawansowane materiały. Największym wyzwaniem podczas renderowania jest utrzymanie stabilnej liczby klatek na sekundę, co bezpośrednio wpływa na komfort użytkownika.

Potok grafiki 3D obejmuje szereg etapów przetwarzania – od transformacji wierzchołków, poprzez rasteryzację, aż po obliczenia shaderów i finalne wyświetlenie. Każdy etap obciąża układ graficzny, a czas jego wykonania rośnie wraz ze złożonością modelu. Liczba trójkątów oraz złożoność shaderów uchodzą za główne determinanty wydajności.

W praktyce użytkownicy oczekują renderowania w przedziale 30-60 FPS. Spadki poniżej 30 FPS zauważalnie obniżają jakość odbioru, a w przypadku headsetów XR mogą prowadzić do dyskomfortu fizycznego.

Współczesne przeglądarki obiektów przestrzennych stanowią złożone systemy renderowania czasu rzeczywistego, działające na urządzeniach

o znacznie różniących się parametrach sprzętowych. Urządzenia mobilne, smartfony oraz autonomiczne zestawy rzeczywistości mieszanej (XR) oferują odmienne możliwości obliczeniowe, architekturę podsystemów graficznych oraz strategie zarządzania energią. Ocena wydajności takich aplikacji wymaga zrozumienia kluczowych czynników wpływających na płynność renderowania, zużycie zasobów oraz ogólny komfort użytkownika. Najważniejsze czynniki determinujące działanie przeglądarek 3D obejmują złożoność geometryczną modeli, strukturę materiałów i tekstur, architekturę GPU i CPU, techniki optymalizacyjne oraz metody pomiaru oraz profilowania (Akenine-Möller, Haines, Hoffman, 2018).

Z punktu widzenia technologii renderowania czasu rzeczywistego, złożoność geometryczna stanowi najważniejszy czynnik wpływający na wydajność przeglądarek obiektów przestrzennych. Modele 3D składają się z trójkątów oraz wierzchołków, które muszą zostać każdorazowo przetworzone podczas wyświetlania sceny. W praktyce to właśnie liczba elementów siatki, ich rozmieszczenie oraz konstrukcja materiałowa wpływają na obciążenie poszczególnych etapów potoku renderowania.

W fazie przetwarzania wierzchołków GPU wykonuje transformacje przestrzenne, obliczenia macierzy oraz przygotowanie danych do etapu rasteryzacji. Każdy dodatkowy wierzchołek zwiększa czas obliczeń w sposób liniowy (Shreiner, Woo, Neider, Davis, 2013). W praktyce oznacza to, że modele zawierające kilkaset tysięcy wierzchołków mogą znacząco wydłużać czas renderowania klatek szczególnie na urządzeniach o ograniczonej mocy GPU, takich jak średniej klasy smartfony.

Rasteryzacja trójkątów jest obciążająca zwłaszcza dla modeli zawierających dużą liczbę niewielkich, gęsto upakowanych elementów siatki. Badania wykazują, że już przekroczenie progu około 100 tysięcy trójkątów zauważalnie obniża płynność renderowania na urządzeniach mobilnych (Azuma i in., 2001; Smith, Brown, 2020). Platformy XR charakteryzują się co prawda większą przepustowością GPU, lecz obciążenie jest dodatkowo zwiększane przez renderowanie stereoskopowe.

Po rasteryzacji każdy fragment (piksel) wymaga obliczeń shaderów odpowiedzialnych za oświetlenie, tekstury oraz efekty materiałowe. Ten etap potoku renderowania jest szczególnie kosztowny w przypadku modeli

z wieloma materiałami oraz złożonymi shaderami PBR (ang. Physically Based Rendering). Na GPU urządzeń mobilnych obliczenia cieniowania mogą odpowiadać nawet za 60% całkowitego czasu renderowania (Akenine-Möller, Haines, Hoffman, 2018).

Liczba elementów geometrycznych nie jest jedynym istotnym parametrem. Znaczącą rolę odgrywa liczba submesh i materiałów przypisanych do jednego modelu. Każdy submesh generuje odrębne wywołanie rysowania (draw call), co tworzy dodatkowy narzut na CPU. Narzut ten wynosi około 0,3-0,7 ms na jedno wywołanie (Kowalski, 2022), co sprawia, że liczba draw calls jest często bardziej ograniczająca niż liczba trójkątów. Dla urządzeń mobilnych zaleca się nieprzekraczanie progu 100-150 wywołań na klatkę (Akenine-Möller, Haines, Hoffman, 2018).

Urządzenia mobilne oraz zestawy XR (np. Meta Quest 3) znacząco różnią się architekturą sprzętową, co wpływa na ich możliwości renderowania modeli 3D. W praktyce prowadzi to do odmiennych profili wydajnościowych nawet przy identycznych aplikacjach i tych samych modelach testowych.

Najnowsze urządzenia XR korzystają z układów projektowanych specjalnie z myślą o aplikacjach przestrzennych. Meta Quest 3, bazujący na Snapdragon XR2 Gen 2, oferuje szereg optymalizacji sprzętowych i programowych ukierunkowanych na renderowanie stereoskopowe i obliczenia graficzne (Wang, Sun, 2018). W praktyce przekłada się to na 2540% wyższą wydajność w renderowaniu modeli 3D w porównaniu ze smartfonami o podobnej mocy obliczeniowej.

Z drugiej strony smartfony mogą przeznaczyć całość zasobów GPU na renderowanie modeli 3D, ponieważ nie wykonują kosztownych obliczeń związanych z mapowaniem i śledzeniem przestrzennym. Dzięki temu osiągają korzystne wyniki w scenariuszach krótkotrwałego obciążenia. Jednakże architektura systemów chłodzenia – w smartfonach znacznie uboższa niż w zestawach XR – sprawia, że ich wydajność podlega szybkiemu throttlingowi. Procesy te absorbują łącznie nawet 25-30% zasobów GPU i CPU, co obniża efektywnie dostępny budżet obliczeniowy dla samego renderowania modeli 3D. Mimo przewagi sprzętowej XR, to dodatkowe obciążenie może prowadzić do sytuacji, w której smartfon osiąga wyższą liczbę FPS przy

prostych modelach, podczas gdy XR zachowuje przewagę przy geometrii bardziej złożonej.

Zgodnie z analizą (Jerald, 2015), modele liczące ponad 250 tysięcy trójkątów zwiększają zużycie energii o 40-60%, co prowadzi do szybkiego podgrzewania układu SoC (ang. System on a Chip). Smartfony zazwyczaj aktywują mechanizmy redukcji taktowania po 5-10 minutach intensywnego renderowania (Buchwald, 2020). Zestawy XR posiadają wydajniejsze systemy odprowadzania ciepła, dzięki czemu mogą utrzymywać stabilną wydajność przez dłuższy czas.

Drugą najważniejszą grupą czynników wpływających na wydajność przeglądarek obiektów przestrzennych są tekstury i materiały. Modele wykorzystywane w nowoczesnych aplikacjach często bazują na materiałach PBR, korzystających z wielu map tekstur.

Tekstury o rozdzielczości 4096×4096 pikseli mogą zajmować znaczną część pamięci dostępnej dla GPU. Przekroczenie pojemności pamięci operacyjnej lub pamięci podręcznej GPU prowadzi do transferów pomiędzy RAM a VRAM, co skutkuje drastycznym spadkiem płynności (Colborne, 2011).

Urządzenia mobilne, dysponujące ograniczoną ilością pamięci i znacznie mniejszą przepustowością magistrali pamięci niż komputery stacjonarne, są szczególnie wrażliwe na duże tekstury i częste zmiany materiałów.

Materiał PBR wymaga intensywnych obliczeń związanych z odbiciami światła, rozproszeniem, mapami normalnych i prawidłową transformacją przestrzeni. Na urządzeniach mobilnych przeciętny model PBR może zużywać nawet ponad połowę czasu renderowania klatki (Akenine-Möller, Haines, Hoffman, 2018).

W obliczu ograniczeń sprzętowych urządzeń mobilnych i XR kluczowe znaczenie mają techniki optymalizacji stosowane w przeglądarkach obiektów przestrzennych. Wskazywane jest kilka metod szczególnie skutecznych w tym kontekście.

- Level of Detail (LOD) – mechanizm LOD polega na przełączaniu pomiędzy różnymi wariantami siatki w zależności od odległości obiektu od kamery. Badania wykazują możliwość redukcji kosztów renderowania nawet o 40-70% przy minimalnej utracie jakości (Akenine-Möller, Haines, Hoffman, 2018).

- Occlusion curling – technika ta pomija renderowanie obiektów przesłoniętych. Jest szczególnie skuteczna w scenach z dużą liczbą modeli 3D, gdzie typowa redukcja przetwarzanej geometrii wynosi 30-50% (Marrs, Shirley, Wald, 2021).
- GPU instancing i batching – łączenie obiektów wykorzystujących ten sam materiał oraz renderowanie wielu kopii tego samego modelu jednocześnie znacząco redukuje narzut CPU, można uzyskać nawet 90% redukcję draw calls oraz wzrost FPS o 30-45% (Cignoni i in., 2008).
- Adaptacyjne renderowanie – nowoczesne przeglądarki 3D monitorują wydajność w czasie rzeczywistym. Algorytmy adaptacyjne zmieniają rozdzielczość renderowania, poziomy LOD oraz intensywność post-processingu 5-10 razy na sekundę, utrzymując stabilną liczbę klatek (Nowak, Wiśniewski, 2021). Rozwiązania tego typu są szczególnie istotne na urządzeniach mobilnych, gdzie obciążenie potrafi gwałtownie wzrosnąć w zależności od modelu 3D.
- Foveated rendering – technika ta, charakterystyczna dla urządzeń XR, redukuje dokładność renderowania w obszarach nieobjętych bezpośrednim skupieniem wzroku użytkownika. W trakcie manipulacji modelami obniżenie jakości w peryferiach może redukować obciążenie GPU nawet o 50-70% (Cummings, Bailenson, 2016).

Ocena wydajności przeglądarek obiektów przestrzennych wymaga stosowania różnorodnych metod łączących monitorowanie czasu rzeczywistego, profilowanie oraz analizę danych po zakończeniu sesji testowej. Do szybkiej diagnozy problemów wykorzystuje się m.in.: nakładki diagnostyczne wyświetlające FPS i obciążenie GPU, znaczniki czasowe mierzące czas wybranych etapów renderowania. Profilery CPU i GPU dostarczają szczegółowych informacji o transformacjach geometrycznych, czasie wykonania shaderów, wykorzystaniu pamięci przez siatki i tekstury, liczbie wywołań rysowania (Colborne, 2011; Kowalski, 2022). Specjalistyczne narzędzia, takie jak Snapdragon Profiler czy Unity GPU Profiler, pozwalają wyodrębnić wąskie gardła renderowania modeli 3D. zależności pomiędzy złożonością modeli a spadkami wydajności (Garcia, Lopez, 2019). Łączenie różnych metod pozwala na kompleksowe zrozumienie zachowania przeglądarki 3D na różnych platformach sprzętowych.

Przeglądarki obiektów przestrzennych działające na urządzeniach mobilnych i XR podlegają złożonym uwarunkowaniom wydajnościowym wynikającym zarówno z parametrów modeli 3D, jak i charakterystyki sprzętu. Najważniejszym czynnikiem determinującym wydajność jest złożoność geometryczna modeli, podczas gdy draw calls, tekstury oraz złożone materiały PBR stanowią dodatkowe istotne obciążenia.

Urządzenia XR dysponują większą mocą obliczeniową, lecz część zasobów przeznaczają na procesy śledzenia przestrzennego, co zmienia ich profil wydajnościowy. Z kolei smartfony osiągają wysoką chwilową wydajność, lecz są podatne na przegrzewanie i throttling. Optymalizacja aplikacji wymaga stosowania technik takich jak LOD, occlusion culling, batching, instancing oraz adaptacyjne renderowanie. Rzetelna ocena wydajności wymaga natomiast połączenia profilowania CPU i GPU, monitorowania czasu rzeczywistego oraz analizy logów po zakończeniu testów. Zestawienie tych czynników tworzy solidną podstawę metodologiczną do badań porównawczych przeglądarek obiektów przestrzennych w środowisku urządzeń mobilnych oraz XR.

Metodyka badań

Celem badań była ocena wydajności przeglądarki modeli 3D działającej na różnych urządzeniach bez zmieniania architektury aplikacyjnej. W tym celu stworzono program testowy w Unity 2022.3 LTS działający w trybie Android 2D, co umożliwiło uruchomienie identycznego kodu na wszystkich platformach. Meta Quest 3 został przetestowany w trybie kompatybilności Android, nie w natywnym VR. To podejście eliminuje różnice implementacyjne między platformami i zapewnia identyczną architekturę aplikacji na każdym urządzeniu testowym.

Testy przeprowadzono na urządzeniach reprezentujących różne generacje wydajności mobilnej, dzięki czemu możliwe było porównanie ich w szerokim zakresie zastosowań (tabela 1). Teoretyczny ranking wydajności GPU według specyfikacji producentów to Adreno 740 > 735 > 660 > 650. Meta Quest 3 posiada najnowszy dedykowany układ XR, POCO F6 wykorzystuje najnowszą mobilną architekturę z 2024 roku, Samsung Galaxy Z Flip 3

reprezentuje wysoką klasę z 2021 roku, a Samsung Galaxy S20 FE stanowi średnią klasę z 2020 roku.

Tabela 1. Specyfikacja techniczna urządzeń testowych

Urządzenie	Model systemowy	Procesor	GPU	RAM
Samsung Galaxy S20 FE	SM-G780G	Snapdragon 865	Adreno 650	6GB
Samsung Galaxy Z Flip 3	SM-F711B	Snapdragon 888	Adreno 660	8GB
Xiaomi POCO F6	24069PC21G	Snapdragon 8s Gen 3	Adreno 735	12GB
Meta Quest 3	Oculus Quest	Snapdragon XR2 Gen 2	Adreno 740	8GB

Źródło: opracowanie własne.

Do testów wykorzystano dziewięć modeli 3D, których złożoność mieściła się w zakresie 12-1349052 trójkątów. Każdy model reprezentował inny poziom szczegółowości, od prostych prymitywów po skomplikowane konstrukcje mechaniczne (tabela 2). Wykorzystano m.in. gotowe modele Stanford oraz Utah Teapot zapewniając porównywalność wyników z istniejącą literaturą naukową w obszarze grafiki komputerowej. Stanford Bunny i Stanford Dragon stanowią uznane benchmarki wydajnościowe stosowane od lat 90 (Turk, Levoy, 1995; Curless, Levoy, 1996).

Tabela 2. Parametry modeli testowych

Model	Trójkąty	Wierzchołki	Kategoria złożoności
Cube	12	24	Bardzo niska
Sphere	768	515	Niska
Cylinder	5 292	10,71	Niska-średnia
Utah_Teapot	17 382	8 706	Średnia
Stanford_Bunny	69 451	34 838	Średnia-wysoka
Wkład_u-rurowy	74 766	36 125	Średnia-wysoka
Zbiornik	277 644	138 106	Wysoka
Stanford_Dragon	869,942	1 384 338	Bardzo wysoka
System_rurowy	1 349 052	666 626	Ultrawysoka

Źródło: opracowanie własne.

Monitorowano następujące metryki: FPS (średni), wykorzystanie CPU, wykorzystanie GPU, zużycie RAM, temperatura urządzenia.

Wyniki eksperymentów

Wydajność FPS

Meta Quest 3 utrzymywał stałe 24 FPS, co wskazuje na systemowy limit odświeżania w trybie Android. Wyniki pomiarów liczby klatek na sekundę (FPS), zestawione w tabeli 3, ukazały różnice pomiędzy badanymi urządzeniami. Meta Quest 3 utrzymywał stały poziom 24,0 FPS, co odpowiada czasowi renderowania pojedynczej klatki wynoszącemu 41,7 ms. Jest to konsekwencja ograniczeń podczas działania w środowisku kompatybilności Android. Smartfony biorące udział w badaniu uzyskiwały średnio około 30 FPS, z czasem renderowania bliskim 33,3 ms. Najwyższą wydajność uzyskał POCO F6, który utrzymywał średnio 30 FPS dla wszystkich modeli. Urządzenia Samsung notowały spadki wydajności po przekroczeniu progu ok. 277 644 trójkątów.

Tabela 3. FPS średnie/min/max dla trzech pomiarów dla różnych urządzeń i modeli

Model	Samsung Galaxy S20 FE	Samsung Galaxy Z Flip 3	Xiaomi POCO F6	Meta Quest 3
Cube	29.9 / 29.9 / 29.9	30.0 / 28.4 / 30.0	29.5 / 28.4 / 30.1	24.0 / 24.0 / 24.0
Sphere	29.9 / 29.9 / 29.9	30.0 / 28.4 / 30.0	30.1 / 30.1 / 30.1	24.0 / 24.0 / 24.0
Cylinder	29.9 / 29.9 / 29.9	30.0 / 28.4 / 30.0	30.1 / 30.1 / 30.1	24.0 / 24.0 / 24.0
Utah_Teapot	29.9 / 29.9 / 29.9	30.0 / 28.4 / 30.0	30.1 / 30.1 / 30.1	24.0 / 24.0 / 24.0
Stanford_Bunny	29.9 / 29.9 / 30.0	30.0 / 28.4 / 30.0	20.2 / 30.1* / 30.1	24.0 / 24.0 / 24.0
Wklad_u-rurowy	29.9 / 29.9 / 29.9	30.6 / 28.4 / 32.6	30.1 / 29.8 / 30.1	24.0 / 24.0 / 24.0
Zbiornik	29.9 / 29.9 / 29.9	30.6 / 28.4 / 32.6	30.1 / 30.0 / 30.1	24.0 / 24.0 / 24.0
Stanford_Dragon	29.7 / 29.0 / 29.9	30.6 / 28.4 / 32.6	29.8 / 28.5 / 30.1	24.0 / 24.0 / 24.0
System_rurowy	29.6 / 29.0 / 30.0	30.6 / 28.4 / 32.6	29.9 / 29.3 / 30.1	23.9 / 23.5 / 24.0

Źródło: opracowanie własne.

Temperatura

Różnice temperatur pomiędzy urządzeniami były znaczące. Meta Quest 3 podczas testów pracował w zakresie 48-58°C, co pokazuje, że dedykowane chłodzenie zoptymalizowane dla aplikacji VR nie zapobiega wzrostowi temperatur w trybie Android. Smartfony natomiast pozostawały znacznie chłodniejsze – POCO F6 osiągał temperatury 31-32°C (jedynie dla modelu

Satnford_Dragon 29,8°C), Galaxy Z Flip 3 około 32°C, a Galaxy S20 FE od 27 do 33°C. W praktyce oznacza to, że zestaw VR był średnio o 50-75% cieplejszy niż urządzenia mobilne. Podwyższona temperatura Meta Quest 3 wynikała przede wszystkim z większego obciążenia procesora w trybie zgodności, podczas gdy smartfony wykazywały wyższą efektywność energetyczną w aplikacjach działających w 2D. Wyniki przedstawiono w tabeli 4.

Tabela 4. Temperatura urządzeń [°C] dla różnych modeli testowych

Model	Samsung Galaxy S20 FE	Samsung Galaxy Z Flip 3	Xiaomi POCO F6	Meta Quest 3
Cube	31.1	30.3	31.7	48.0
Sphere	31.1	30.3	31.7	48.0
Cylinder	31.1	30.3	31.8	48.0
Utah_Teapot	31.1	30.3	31.9	48.0
Stanford_Bunny	31.1	30.3	31.9	58.0
Wkład_u-rurowy	28.9	30.5	32.2	55.0
Zbiornik	28.9	30.5	32.3	53.0
Stanford_Dragon	28.9	30.5	29.8	48.0
System_rurowy	29.4	30.5	32.1	48.5

Źródło: opracowanie własne.

Zużycie zasobów

Zależność pomiędzy obciążeniem GPU a złożonością renderowanych modeli jest zaprezentowana w tabeli 5. Meta Quest 3 wykazywał wzrastające obciążenie GPU: 0,2% dla Cube, 11,5% dla Sphere, 79,4% dla Cylinder, a następnie stałe 100% dla modeli od Utah_Teapot wzwyż. Wzorzec obciążenia był identyczny z urządzeniami mobilnymi.

Urządzenia mobilne wykazywały obciążenie GPU od około 0,2% przy najprostszych modelach do 100% w przypadku obiektów bardziej złożonych. Kluczowy punkt przecięcia zidentyfikowano dla modelu Utah_Teapot (17 382 trójkąty). Proste modele, takie jak Cube, Sphere czy Cylinder, nie wywoływały pełnego obciążenia GPU, natomiast obiekty o średniej i dużej szczegółowości praktycznie zawsze osiągały maksymalny poziom wykorzystania zasobów.

Tabela 5. Wykorzystanie GPU [%] dla różnych urządzeń i modeli

Model	Samsung Galaxy S20 FE	Samsung Galaxy Z Flip 3	Xiaomi POCO F6	Meta Quest 3
Cube	0.2	0.2	0.2	0.2
Sphere	11.5	11.5	11.5	11.5
Cylinder	79.4	79.4	79.4	79.4
Utah_Teapot	100.0	100.0	100.0	100.0
Stanford_Bunny	100.0	100.0	100.0	100.0
Wklad_u-rurowy	100.0	100.0	100.0	100.0
Zbiornik	100.0	100.0	100.0	100.0
Stanford_Dragon	100.0	100.0	100.0	100.0
System_rurowy	100.0	100.0	100.0	100.0

Źródło: opracowanie własne.

Zużycie pamięci operacyjnej dla różnych urządzeń i modeli testowych przedstawione jest w tabeli 6. Meta Quest 3 okazuje się najbardziej efektywny pod względem wykorzystania pamięci. Smartfony Samsung zachowują się przewidywalnie, a ich zużycie pamięci rośnie w sposób proporcjonalny do złożoności modeli. POCO F6 wykazywał najwyższe bazowe zużycie RAM (21-22 MB) dla prostych modeli. Pozostałe urządzenia charakteryzowały się około 6 MB bazowo. Meta Quest 3 utrzymywał najbardziej stabilne zarządzanie pamięcią z minimalnymi wahaniami w zakresie 5,3-5,4 MB dla większości modeli.

Tabela 6. Zużycie pamięci operacyjnej [MB] dla różnych urządzeń i modeli

Model	Samsung Galaxy Z Flip 3	Samsung Galaxy S20 FE	Xiaomi POCO F6	Meta Quest 3
Cube	5.5-5.6	5.7-6.1	5.5-21.6	5.4
Sphere	5.6-6.4	5.6-6.6	5.8-22.7	5.4
Cylinder	6.4-8.2	7.3-8.6	7.3-8.6	5.4
Utah_Teapot	7.8-8.2	7.4-8.7	7.4-8.7	5.4
Stanford_Bunny	7.8-8.2	6.7-8.6	7.6-8.5	5.3-5.4
Wklad_u-rurowy	5.5-5.7	5.9-6.3	5.7	5.3-5.4
Zbiornik	8.9-14.1	9.4-14.1	8.9	5.5-8.6
Stanford_Dragon	15.5-19.8	15.6-20.8	15.6-22.6	5.4
System_rurowy	5.5-21.0	5.7-21.7	5.7-21.2	5.4-20.8

Źródło: opracowanie własne.

Podsumowanie

Badania wydajności przeglądarki obiektów przestrzennych przeprowadzono dla czterech urządzeń testowych, dziewięciu modeli testowych o złożoności od 12 do 1 349 052 trójkątów, po trzy powtórzenia każdej sesji. Eksperymenty umożliwiły analizę wpływu geometrii na wydajność. Wyniki potwierdziły zależność wydajności od możliwości sprzętowych urządzeń. Ranking średniego FPS pokazał, że wszystkie urządzenia mobilne osiągnęły zbliżone rezultaty: Xiaomi POCO F6 – 30.0 FPS, Samsung Galaxy Z Flip 3 – 29.9 FPS, Samsung Galaxy S20 FE – 29.8 FPS przewyższające Meta Quest 3, który uzyskał najniższy wynik (24.0 FPS) z powodu systemowego limitu narzuconego przez tryb kompatybilności Android 2D. Jednocześnie teoretyczna analiza generacji GPU wskazuje, że Meta Quest 3 dysponuje najnowocześniejszą architekturą (Adreno 740), lecz jej potencjał pozostaje niewykorzystany.

Meta Quest 3, mimo zaawansowanej konstrukcji i aktywnego chłodzenia, pracował ze stałą częstotliwością 24 FPS niezależnie od złożoności modeli. Wysoka temperatura pracy (średnio 52°C) oraz stałe wykorzystanie GPU na poziomie 100% wskazują na ograniczenia wynikające z warstwy kompatybilności Android, a nie ze specyfiki sprzętu. Pełny potencjał urządzenia może zostać ujawniony wyłącznie w aplikacjach natywnych VR/XR.

Wpływ złożoności geometrycznej modeli na wydajność najsilniejszy jest dla urządzeń mobilnych. Wzrost liczby trójkątów prowadzi do systematycznych spadków FPS. Zidentyfikowano dwa kluczowe progi wydajnościowe:

- 277 644 trójkątów, do którego wszystkie urządzenia mobilne utrzymywały stałe ~30 FPS,
- 869 942 trójkątów, przy którym pojawiały się pierwsze spadki wydajności (np. POCO F6: 29.8 FPS, S20 FE: 29.7 FPS).

Najbardziej wydajne urządzenie – Xiaomi POCO F6 – utrzymało ponad 30 FPS aż do modelu Stanford_Dragon (869 942 trójkątów) i jako jedyne obsłużyło cały zestaw dziewięciu modeli, z minimalnym spadkiem do 29.3 FPS przy najcięższym modelu System_ruowy (1 349 052 trójkątów).

Wyniki dotyczące pamięci RAM wykazały liniowy wzrost zużycia wraz ze złożonością geometrii. Najprostszy model Cube wymagał około 6 MB

pamięci, natomiast Stanford_Dragon 15.5-22.6 MB. Najwyższe wykorzystanie pamięci dla złożonych modeli odnotowane na POCO F6 (12 GB RAM) przy najwyższej ilości średnich FPS, potwierdza bezpośrednią zależność między zasobami urządzenia a wydajnością i stabilnością pracy.

Wydajność przeglądarki obiektów przestrzennych jest silnie zależna od generacji GPU, złożoności geometrii modeli oraz ograniczeń systemowych platformy. Wyniki dostarczają pewnych wskazań projektowych: urządzenia z układami Adreno 660 i starszymi wymagają ograniczania złożoności modeli do ok. 277 644 trójkątów lub stosowania LOD; urządzenia nowsze (Adreno 735/740) mogą obsługiwać modele do ok. 870 000 trójkątów przy zachowaniu pełnej płynności. Badania tworzą solidną podstawę dla dalszych prac nad optymalizacją renderowania w mobilnych i mieszanych aplikacjach przestrzennych.

Literatura

1. Akenine-Möller, T., Haines, E., Hoffman, N. (2018). *Real-Time Rendering* (4th ed.). A K Peters/CRC Press.
2. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6), 34-47. <https://doi.org/10.1109/38.963459>
3. Bailenson, J. (2018). *Experience on demand: What virtual reality is, how it works, and what it can do*. W.W. Norton & Company.
4. Blascovich, J., Bailenson, J. (2011). *Infinite Reality: Avatars, Eternal Life, New Worlds, and the Dawn of the Virtual Revolution*. William Morrow.
5. Bowman, D. A., McMahan, R. P. (2007). Virtual Reality: How Much Immersion Is Enough? *Computer*, 40(7), 36-43.
6. Buchwald, P. (2020). *Urządzenia mobilne w systemach rzeczywistości wirtualnej*. Wydawnictwo Naukowe PWN.
7. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G. (2008). MeshLab: an Open-Source Mesh Processing Tool. *Sixth Eurographics Italian Chapter Conference*, 129-136.
8. Colborne, G. (2011). *Prostota i użyteczność. Projektowanie rozwiązań internetowych, mobilnych i interaktywnych*. Helion.

9. Cummings, J. J., Bailenson, J. N. (2016). How immersive is enough? A meta-analysis of the effect of immersive technology on user presence. *Media Psychology*, 19(2), 272-309.
10. Curless, B., Levoy, M. (1996). A volumetric method for building complex models from range images. *SIGGRAPH '96: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, 303-312. <https://doi.org/10.1145/237170.237269>
11. Garcia, F., Lopez, D. (2019). Mobile VR: Performance Analysis of 3D Rendering Techniques. *Mobile Information Systems*, 2019, 123456.
12. Jerald, J. (2015). *The VR Book: Human-Centered Design for Virtual Reality*. Association for Computing Machinery & Morgan & Claypool.
13. Kowalski, M. (2022). Analiza wydajności aplikacji VR na urządzeniach mobilnych. *Informatyka*, 45(3), 123-135.
14. Marrs, A., Shirley, P., Wald, I. (red.). (2021). *Ray Tracing Gems II: Next Generation Real-Time Rendering with DXR, Vulkan, and OptiX*. Apress. <https://doi.org/10.1007/978-1-4842-7185-8>
15. Nowak, A., Wiśniewski, P. (2021). Optymalizacja renderowania grafiki 3D w aplikacjach mobilnych. *Technologie Informacyjne*, 12(2), 98-110.
16. Sawa, J. (2025). *Ocena wydajności przeglądarki obiektów przestrzennych na wybranych urządzeniach rzeczywistości mieszanej i mobilnych*. Praca magisterska. Akademia Bialska im. Jana Pawła II.
17. Shreiner, D., Woo, M., Neider, J., Davis, T. (2013). *OpenGL Programming Guide: The Official Guide to Learning OpenGL* (8th ed.). Addison-Wesley Professional.
18. Smith, J., Brown, L. (2020). Performance Evaluation of 3D Object Viewers on VR Devices. *Journal of Virtual Reality and Broadcasting*, 17(4), 45-59.
19. Turk, G., Levoy, M. (1994). Zippered polygon meshes from range images. *SIGGRAPH '94: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, 311-318. <https://doi.org/10.1145/192161.192241>
20. Wang, P., Sun, Y. (2018). Optimization Strategies for 3D Graphics Rendering on Mobile VR Platforms. *IEEE Transactions on Visualization and Computer Graphics*, 24(7), 2105-2114.

Netografia

1. Unity Technologies. (2022). *Universal Render Pipeline Documentation Unity 2022.3 LTS*. Pobrane z: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal> (data dostępu: 12.10.2025).