

Porównanie wydajności trójwymiarowych gier z użyciem silników CryEngine i Unity

Hubert Żukowski*

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W artykule przedstawiono wyniki badań wydajności aplikacji utworzonych na silnikach gier wideo: Unity i CryEngine. Przeprowadzone badania skupiały się na zestawieniu dwóch aplikacji utworzonych na wskazanych silnikach. Wybrano parametry, dla których zostało wykonane porównanie, mianowicie liczbę klatek na sekundę, wykorzystanie procesora, zużycie pamięci RAM, a także czas generowania obiektów 3D przez aplikacje. Wykonane aplikacje korzystały z tych samych zasobów i zbliżonego kodu. Postawiona w artykule hipoteza – Unity jest silnikiem wydajniejszym niż CryEngine – została zweryfikowana i częściowo udowodniona: oba silniki są wydajniejsze w zależności od sposobu wykorzystania.

Słowa kluczowe: unity; cryengine; silnik gier; wydajność

*Autor do korespondencji.

Adres e-mail: h.zukowski@outlook.com

Comparison of 3D games' efficiency with use of CRYENGINE and Unity game engines

Hubert Żukowski*

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

Abstract. This article presents the results of the performance studies of applications created with use of CryEngine and Unity game engines. Presented research was mainly focused on comparison of two applications created on selected engines. Several parameters were selected for the research: frame rate, CPU usage, RAM usage and generation time of 3D objects. Created applications were built with the use of the same graphic resources and similar source code. The hypotheses set in the article – Unity game engine is more efficient than CryEngine – have been verified and partially confirmed: engines are more efficient in different environments.

Keywords: unity; cryengine; game engine; performance

*Corresponding author.

E-mail address: h.zukowski@outlook.com

1. Wstęp

Branża gier wideo jest jedną z najszybciej rozwijających się dziedzin z pogranicza informatyki. Od 2017 nieprzerwanie dominuje nad rynkiem muzycznym, filmowym, a także książkowym [11]. W roku 2018 wartość rynku gier wideo wzrosła niemal o 27% do wartości 135 mld dolarów [1].

Koncerny specjalizujące się w produkcji gier odnoszą sukcesy od początku lat 70, a roczna wartość ich przychodów szacowana jest na ponad 100 mld dolarów [13]. Nie może więc dziwić fakt, że każdego roku pojawia się coraz więcej firm zajmujących się tworzeniem tego typu oprogramowania.

Jednym z najpopularniejszych i najszerzych podziałów gier wideo jest pogrupowanie ze względu na platformy, na których są dostępne. Wobec tego najczęściej mowa o grach PC, grach mobilnych oraz grach konsolowych [5].

Gry wideo są interaktywnym oprogramowaniem komputerowym. Ich celem jest najczęściej rozrywka bądź edukacja wymagająca od użytkownika rozwiązywania zadań logicznych i/lub zręcznościowych [5], a ich początki sięgają końca pierwszej połowy XX wieku. Uruchomienie gry wiąże

się z wykorzystaniem zasobów komputera. Nowoczesne gry wymagają coraz potężniejszych kart graficznych, procesorów, czy też większej ilości pamięci RAM, przez co rynek gier wideo napędza także producentów podzespołów, czy samych komputerów, laptopów, konsol i smartfonów.

Dzisiejsze gry są tworzone za pomocą silników gier [5] projektowanych w oparciu o sprecyzowany paradygmat obiektowy i wykorzystywanych przez przedsiębiorstwa, które produkują gry wideo zarówno na komputery personalne, smartfony, jak i konsole. Aktualne rozwiązania w dziedzinie produkcji gier opierają się o dwa typy silników: tworzone od podstaw przez wytwórnię gier oraz wykorzystanie istniejących rozwiązań, wśród których najpopularniejsze są Unreal Engine, Unity oraz CryEngine [10]. Silniki te oferują pełny zestaw funkcji niezbędnych do zbudowania interaktywnej aplikacji 3D, tj.:

- silnik renderowania – moduł graficzny pozwalający na tworzenie obiektów 3D oraz efektów wizualnych;
- silnik fizyki – moduł odpowiedzialny za zjawiska fizyczne występujące w grze: kolizje, grawitację oraz reakcje obiektów 3D;
- silnik audio – umożliwiający odtwarzanie efektów dźwiękowych oraz muzyki;

- silnik sieciowy – moduł pozwalający na zaprogramowanie trybów wieloosobowych rozgrywanych w czasie rzeczywistym poprzez wymianę danych pomiędzy urządzeniami graczy.

W badaniach przeprowadzonych na platformie Steam najczęściej wykorzystywanym silnikiem gier jest Unreal Engine, Unity, id Tech, Source oraz CryEngine (tabela 1.). Badania te dotyczą gier VR, 3D oraz 2D, czyli wszystkich dostępnych rodzajów na platformie. W badaniu wzięto pod uwagę ponad 50 tys. tytułów [6].

Tabela 1. Wykorzystanie silników gier na platformie Steam [6]

Silnik gier	Wykorzystanie
Unreal Engine	23%
Unity	11%
id Tech	6%
Source	4%
CryEngine	3%
Inne (GameMaker, Fox, Frostbite...)	53%

Najpopularniejsze z wykorzystywanych silników umożliwiają produkcję gier na wiele platform jednocześnie. Dzięki temu producenci mogą zaoszczędzić na przenoszeniu gier na inne platformy.

W niniejszej pracy postawiona została następująca hipoteza: silnik Unity jest wydajniejszy niż CryEngine.

Do potwierdzenia przedstawionej tezy przeprowadzona została analiza porównawcza aplikacji wykonanych na obu silnikach. Zestawione gry testowe będą wykorzystywać takie same modele 3D oraz skrypty. Silniki Unity i CryEngine wybrane zostały z powodu podobieństwa wpieranych języków i platform (tabela 2.). Oba pozwalają na programowanie w języku C#, a wykonane z ich pomocą aplikacje działają na systemie Windows 10. Kolejnym podobieństwem jest wspieranie gier 3D oraz aplikacji Virtual Reality. Powodem wyboru CryEngine – jako konkurenta Unity – jest brak badań wydajnościowych tego silnika.

Tabela 2. Porównanie cech wybranych silników [3] [4]

Cecha	CryEngine	Unity
Wspierane platformy	Oculus Rift, Windows PC, Xbox One, Playstation 4	iOs, Android, Windows, Mac, Linux, WebGL, Playstation 4, Xbox One, Nintendo 3DS, Oculus Rift, Nintendo Switch
Silnik fizyki	CryPhysics	PhysX
Wspierany język	C#, C++	C#, UnityScript, JavaScript, Boo
Gry 2D	Nie	Tak
Gry 3D	Tak	Tak
VR	Tak	Tak

2. Przegląd literatury

Dostępne materiały badawcze przedstawiają silniki gier wideo nie tylko jako narzędzia do wytwarzania gier, lecz także wskazują na ich użyteczność w środowiskach naukowych. Silniki gier wykorzystuje się w dziedzinach z pogranicza medycyny, architektury [14], meteorologii [8], geologii [2] oraz edukacji [12].

Znaczna część publikacji skupia się na rozwoju silników gier wideo, sięgając do ich historii, niemniej w związku z ciągłymi postępami dziedziny rzadko można spotkać aktualną publikację dotyczącą aktywnie wykorzystywanych silników na obecnym rynku. Podobnie jest z porównywaniem silników.

Na podstawie pracy I. Pachoulakis i G. Pontikakis można dowiedzieć się o sposobie porównania gier bazujących na silnikach Unity i Unreal Engine [9], lecz brakuje podobnych badań dla CryEngine. W pracy [7] poruszono problem porównywania różnych silników, co umożliwiło utworzenie metody badania w niniejszym artykule.

Wobec powyższych argumentów wysunąć można wniosek, że badanie możliwości silników gier wideo ogranicza się do porównania ich zastosowań, a w nielicznych badaniach skupia się na ewaluacji ich wydajności, biorąc się pod uwagę gry różnych wydawców (opracowane przy odmiennych zasobach).

Niniejsza praca porusza więc problem porównywania silników gier wideo poprzez utworzenie – na wybranych środowiskach – aplikacji korzystających z takich samych zasobów (modeli 3D i skryptów).

3. Metoda badań

Na potrzeby badań utworzono dwie aplikacje w oparciu o silniki CryEngine i Unity. Następnie w obu grach odwzorowano takie same poziomy rozgrywki oraz w miarę możliwości utworzono podobne skrypty zarządzające generowaniem obiektów.

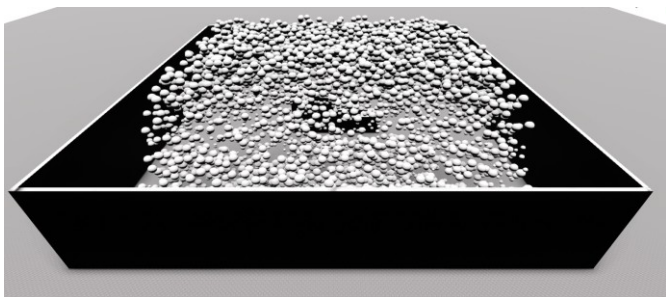
Stworzone w ten sposób aplikacje pozwoliły na stworzenie danej liczby obiektów, a w konsekwencji obciążenie aplikacji i stanowiska badawczego (tabela 3.), a także uzyskanie parametrów pamięci RAM, maksymalnego wykorzystania CPU, klatek na sekundę oraz czasu generowania obiektów przez silnik.

Tabela 3. Parametry stanowiska badawczego

Element	Wersja
System	Windows 10 Home Edition
Procesor	Intel Core i7 4710HQ
Pamięć RAM	8 GB
GPU	GeForce GTX860M
Dysk	SSD, Crucial BX500
Rozdzielczość	1920*1080

Wykonane aplikacje (rys 1.) pozwoliły na generowanie określonej liczby obiektów. Na potrzeby badań wybrano cztery poziomy obciążenia silników: 1000, 5000, 7500 oraz

10000 obiektów. Odczytów wybranych parametrów dokonano po 60 sekundach od wygenerowania wszystkich obiektów. Na wszystkich poziomach testów pomiary powtórzone trzykrotnie dla każdej aplikacji.



Rys. 1. Generator obiektów wykonany na silniku CryEngine

Obiektom generowanym przez aplikację nadano właściwości fizyczne: masę, podatność na grawitację oraz kolizje. Każdy obiekt rozmieszczono losowo (pod względem wysokości i położenia) w obrębie stworzonej sceny 3D.

4. Wyniki badań

Dla pierwszego testu (tabela 4.) przy minimalnym przyjętym obciążeniu – 1000 obiektów – lepsze rezultaty osiągała aplikacja zbudowana na Unity.

Tabela 4. Wyniki pomiarów dla obciążenia 1000 obiektów

CryEngine				
	FPS	CPU	RAM [MB]	Czas gen. [ms]
Badanie 1.	55	55,63%	687,3	104
Badanie 2.	53	55,67%	688,1	122
Badanie 3.	49	53%	687	101
Średnia	52,34	54,77%	687,47	109
Odchyl. std.	3,06	1,5%	0,57	11,36
Unity				
	FPS	CPU	RAM [MB]	Czas gen. [ms]
Badanie 1.	60	19,36%	243,7	53
Badanie 2.	60	19,96%	243,6	81
Badanie 3.	60	37,13%	243,6	87
Średnia	60	25,48%	243,63	73,67
Odchyl. std.	0	10,1%	0,06	18,15

W drugim teście (tabela 5.), dla 5000 wygenerowanych obiektów, rezultaty obu silników zbliżyły się, choć Unity osiągnęło lepsze wyniki w każdym z odczytywanych parametrów.

Wyniki dla trzeciego poziomu (7500 obiektów) badań zaprezentowano w tabeli 6. Na ich podstawie można stwierdzić, że silnik Unity gorzej poradził sobie tylko w przypadku liczby wyświetlanych klatek na sekundę. CryEngine osiągnął słabsze rezultaty w zestawieniu użycia pamięci RAM oraz czasu generowania obiektów.

Przy maksymalnym przyjętym obciążeniu (10000 obiektów) liczba klatek na sekundę w CryEngine nie spadła poniżej granicy płynności obrazu (30 FPS), ale silnik fizyczny CryPhysics nie był w stanie obsłużyć tak dużej liczby

obiektów. Przeciążenie silnika fizyki nie miało wpływu na renderowaną grafikę, a jego zachowanie nie było obiektem badań, wobec czego badanie mogło być kontynuowane. W Unity wartości FPS wyniosły średnio 7 klatek na sekundę. Obciążenie CPU było zbliżone w przypadku obu aplikacji. Wartości czasu generowania obiektów oraz wykorzystania pamięci RAM przemawiają zdecydowanie na korzyść Unity. Wyniki przedstawiono w tabeli 7.

Tabela 5. Wyniki pomiarów dla obciążenia 5000 obiektów

CryEngine				
	FPS	CPU	RAM [MB]	Czas gen. [ms]
Badanie 1.	41	55,28%	688	385
Badanie 2.	43	55,30%	687	339
Badanie 3.	42	55,80%	687,4	360
Średnia	42	55,46%	687,47	361,34
Odchyl. std.	1	0,3%	0,5	23,03
Unity				
	FPS	CPU	RAM [MB]	Czas gen. [ms]
Badanie 1.	43	46,80%	243,5	157
Badanie 2.	39	46,40%	243,6	129
Badanie 3.	45	46,71%	243,6	149
Średnia	42,34	46,64%	243,57	145
Odchyl. std.	3,06	0,21%	0,06	14,42

Tabela 6. Wyniki pomiarów dla obciążenia 7500 obiektów

CryEngine				
	FPS	CPU	RAM [MB]	Czas gen. [ms]
Badanie 1.	34	55,46%	687,9	518
Badanie 2.	37	52,66%	687	593
Badanie 3.	37	57,29%	687	673
Średnia	36	55,14%	687,3	594,67
Odchyl. std.	1,73	2,3%	0,52	77,51
Unity				
	FPS	CPU	RAM [MB]	Czas gen. [ms]
Badanie 1.	17	51,97%	243,9	207
Badanie 2.	16	51,97%	243,9	168
Badanie 3.	17	52,23%	243,5	189
Średnia	16,67	52,06%	243,77	188
Odchyl. std.	0,58	1,5%	0,23	19,52

Tabela 7. Wyniki pomiarów dla obciążenia 10000 obiektów

CryEngine				
	FPS	CPU	RAM [MB]	Czas gen. [ms]
Badanie 1.	31	55,91%	687,9	698
Badanie 2.	28	55,64%	687,4	653
Badanie 3.	33	55,52%	687,5	802
Średnia	30,67	55,69%	687,6	717,67
Odchyl. std.	2,52	0,2%	0,26	76,42
Unity				
	FPS	CPU	RAM [MB]	Czas gen. [ms]
Badanie 1.	7	54,03%	243,9	207
Badanie 2.	7	55,31%	243,6	196
Badanie 3.	8	54,06%	243,8	205
Średnia	7,34	54,47%	243,77	202,67
Odchyl. std.	0,58	0,7%	0,15	5,86

5. Dyskusja wyników

Przy minimalnym i średnim obciążeniu wygenerowanymi obiektami (1000 i 5000) silnik Unity wypadł lepiej we wszystkich parametrach. W każdym z przeprowadzonych testów CryEngine potrzebował ponad 440 MB pamięci RAM więcej, ponadto czas generowania obiektów jest niższy w Unity.

W testach o wyższej liczbie wygenerowanych obiektów CryEngine osiągnął lepsze rezultaty tylko w parametrze klatek na sekundę. Wyniki te nie odzwierciedlają jednak rzeczywistej sprawności silnika, który przy maksymalnym obciążeniu nie radził sobie z obsługą fizyki wszystkich obiektów.

6. Wnioski

Przeprowadzone badania pozwoliły na sformułowanie następujących wniosków:

- Unity potrzebuje mniejszej ilości zasobów do sprawnego działania aplikacji;
- niezależnie od liczby obiektów w kadrze kamery wymagania względem zasobów silnika CryEngine są zbliżone;
- silnik fizyki CryPhysics nie radzi sobie z dużą liczbą obiektów fizycznych.

Wobec wyników badań potwierdzona zostaje teza postawiona we wcześniejszej części artykułu. Silnik Unity jest wydajniejszy w większości testów i bezkonkurencyjny podczas badań na mniejszym obciążeniu aplikacji. Unity w większym stopniu reaguje na zmiany zachodzące w poziomie gry i dostosowuje obciążenie procesora do konkretnych potrzeb, podczas gdy CryEngine stale wymaga takich samych zasobów.

Literatura

- [1] Batchelor J., Global games market value rising to \$134.9bn in 2018, 2018, <https://www.gamesindustry.biz/articles/2018-12-18-global-games-market-value-rose-to-usd134-9bn-in-2018> [10.06.2019]
- [2] Bellanca J. L., Developing a Virtual Reality Environment for Mining Research, 2019, Mining, Metallurgy & Exploration

- [3] Cechy Unity, <https://unity3d.com/unity> [29-06-2019]
- [4] CRYENGINE Feature Index, <https://www.cryengine.com/features> [03-03-2019]
- [5] Gra komputerowa, https://pl.wikipedia.org/wiki/Gra_komputerowa [10.06.2019]
- [6] I researched the market share of game engines on Steam, from over 60,000 Steam games., https://www.reddit.com/r/gamedev/comments/8s20qp/i_researched_the_market_share_of_game_engines_on/ [04.05.2019]
- [7] Mishra P., Shrawankar U., Comparison between Famous Game Engines and Eminent Games, 2016 International Journal of Interactive Multimedia and Artificial Intelligence
- [8] Nowak Ł., Bąk A., Czajkowski T., Wojciechowski K., Modeling and Rendering of Volumetric Clouds in Real-Time with Unreal Engine 4, Computer Vision and Graphics
- [9] Pachoulakis I., Pontikakis G., Combining features of the Unreal and Unity Game Engines to hone development skills, 2015 9th International Conference on New Horizons in Industry, Business and Education
- [10] Popularne silniki graficzne - którego warto się nauczyć?, <https://www.komputerswiat.pl/gamezilla/artykuly/popularne-silniki-graficzne-ktorego-warto-sie-nauczyc/34fhly1> [12.06.2019]
- [11] Radzewicz S., Ubisoft przypomina, że rynek gier już dawno zjadł filmy, muzykę i książki. Infografika nie pozostawia złudzeń, <https://www.spidersweb.pl/rozrywka/2018/02/13/rynek-gier-wideo-2017/> [1.09.2019]
- [12] Salomão A., Andaló F., Vieira M. L. H., How Popular Game Engine Is Helping Improving Academic Research: The DesignLab Case, AHFE 2018: Advances in Human Factors in Wearable Technologies and Game Design
- [13] Świat kupuje coraz więcej gier komputerowych. Sprzedaż sięgnie 98 mld dol. Rocznie, <https://www.forbes.pl/wiadomosci/swiat-kupuje-coraz-wiecej-gier-komputerowych-sprzedaz-siegnie-98-mld-dol-rocznie/v0t1s56> [10.06.2019]
- [14] Wang S., Mao Z., Zeng Ch., Gong H., Li S., Chen B., A New Method of Virtual Reality Based on Unity3D, 2010 18th International Conference on Geoinformatics