

# Comparative analysis of the performance of the Flax engine and Unity

## Analiza porównawcza wydajności silników Flax engine i Unity

Wojciech Szelug\*

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

### Abstract

This article compares the two game engines Unity and Flax engine. The research consisted in comparing the efficiency of individual engines using two similar applications. The first one generates from several hundred to several thousand physical objects and examines the efficiency of engines in terms of physics simulations. The second application generates a forest area filled with the same tree model. The application aims to study the behavior of the engine for various graphics settings. In both applications, parameters such as the number of frames per second, consumption of the processor, graphics card and RAM were tested. Research shows that Unity is more efficient than Flax Engine.

*Keywords:* Unity; Flax Engine; game engine

### Streszczenie

W niniejszym artykule porównane zostały dwa silniki gier Unity i Flax engine. Badania polegały na porównaniu wydajności poszczególnych silników za pomocą podobnych do siebie dwóch aplikacji. Pierwsza z nich generuje od kilkuset do kilku tysięcy modeli 3D kul i bada wydajność silników pod względem symulacji fizyki. Druga aplikacja generuje teren lasu wypełniony takim samym modelem drzewa. Aplikacja ma na celu zbadanie zachowania silnika dla różnych ustawień graficznych. W obu aplikacjach badane zostały takie parametry jak: liczba klatek na sekundę, zużycie procesora, karty graficznej i RAM. Z badań wynika, że Unity jest wydajniejszy od Flax Engine.

*Słowa kluczowe:* Unity; Flax Engine; silniki gier

\*Corresponding author

*Email address:* [Wojciech.Szelug@gmail.com](mailto:Wojciech.Szelug@gmail.com) (W. Szelug)

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

## 1. Wstęp

Od 2017r. na platformie Steam publikowanych jest ponad pięćset gier miesięcznie w tym głównie gry niezależne (ang. indie) [1] a najczęściej używanymi silnikami są Unity, Unreal i GameMaker (Tabela 1). Silnik Flax Engine nie został wymieniony na tej stronie, jednak trzeba brać pod uwagę to, iż widnieje tam komunikat o tym, że te dane mogą być niedokładne

Tabela 1: Popularność silników gier na Steam dla 42163 gier

Silnik	Liczba użyć	Procent użyć
Unity	27233	65%
Unreal	6895	16%
GameMaker	2822	7%
Pozostałe	5 213	12%

Flax Engine został wydany w pełnej wersji w roku 2020 przez Wojciecha Figata [2]. Celem Flax jest stworzenie silnika do gier podobnego do Unity, Unreal engine i CryEngine oraz do silników open-source takich jak Godot, Banshee, Lumix. Obecnie najnowsza wersja silnika to 1.3, ale ma być on dalej rozwijany.

Unity jest jednym z najpopularniejszych silników gier na rynku [3] i posiada ogromną społeczność, która nieustannie tworzy poradniki i udziela się na forach i serwerach Discorda. Mimo że silniki Flax Engine

i Unity są darmowe, to tylko do pewnego progu zarobku. W wersji osobistej można korzystać z narzędzia za darmo do momentu osiągnięcia przychodów w wysokości stu tysięcy dolarów rocznie dla Unity [4] i dwudziestu pięciu tysięcy dolarów przychodów na kwartał dla Flax [5].

W tym artykule porównane zostały wyżej wymienione silniki gier na dwa sposoby. Pierwszy polegał na stworzeniu podobnych aplikacji, w której wygenerowano określoną liczbę obiektów w kształcie kuli i porównano liczbę klatek na sekundę, średnie zużycie wątku procesora, średnie zużycie karty graficznej i średnie zużycie pamięci RAM w MB. Drugi sposób polegał na stworzeniu ogromnego terenu lasu z różnym dystansem rysowania cieni, dla każdego silnika i sprawdzeniu tych samych parametrów co wcześniej. Oba rodzaje aplikacji zostały stworzone z tych samych obiektów 3D i za pomocą podobnych skryptów napisanych w języku C#.

Artykuł został napisany by sprawdzić, jak radzi sobie Flax Engine w porównaniu z Unity. Flax jest nowym silnikiem i nie został jeszcze przetestowany w ten sposób.

## 2. Przegląd literatury

Każdy silnik został zaprojektowany osobno a to znaczy, że istnieją różnice między nimi. Najważniejsze różnice

to wydajność poszczególnych silników. Można znaleźć kilka artykułów, w których porównywane są parametry wydajnościowe silników gier. Najczęściej porównywany jest Unity z innym również popularnym silnikiem. W artykule Huberta Żukowskiego [6] Unity jest porównywany z CryEngine. Autor stworzył podobne do siebie aplikacje, w których wygenerował kilka tysięcy kul. Z badań wynika, że Unity jest bardziej wydajny z mniejszą liczbą obiektów. Gdy obiektów jest dziesięć tysięcy to CryEngine radzi sobie lepiej, ale Unity nadal zużywa mniej zasobów komputera.

W artykule pod tytułem “Porównanie wydajności silników gier na wybranych platformach” [7], autor porównuje Unity oraz Unreal Engine przy pomocy stworzonych przez siebie podobnych gier wirtualnej rzeczywistości na urządzenia mobilne. Z jego badań wynika, że Unity jest nieznacznie wydajniejsze od Unreal Engine.

Andrzej Barczak i Hubert Woźniak [8] porównują Unity, CryEngine i Unreal Engine, lecz w ich badaniach wykorzystali aplikacje w których stworzyli teren lasu a następnie porównywali czas tworzenia się pojedynczej klatki dla poszczególnych silników. Ich wyniki wskazują, że Unity robi to najszybciej w prawie każdym przypadku.

W artykule autorów Eleftheria Christopoulou, Stelios Xinogalos [9] stworzone zostały gry mobilne przy pomocy silników Unity i Unreal Engine 4. Ten artykuł miał na celu sprawdzić, który silnik nadaje się lepiej do tego typu zadań. Tutaj znowu wykazano, że Unity jest lepszy, ale ze względu na większą liczbę poradników i większą społeczność. Sprawdzono też wielkość aplikacji przed i po zbudowaniu z czego wynikało, że Unity zmniejszyło rozmiar aplikacji a Unreal zwiększyło.

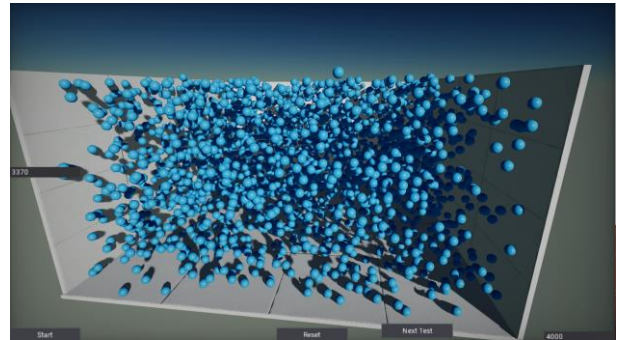
W ostatnim artykule [10] porównano Unity i Unreal Engine w aspekcie tworzenia wirtualnych pokazów modeli pochodzących ze skanowania 3D. Ponownie stwierdzono, że to Unity jest wydajniejsze pod każdym względem.

Patrząc na wyżej wymienione badania można postawić tezę, że silnik Unity jest bardziej wydajny niż Flax Engine

### 3. Metoda badań

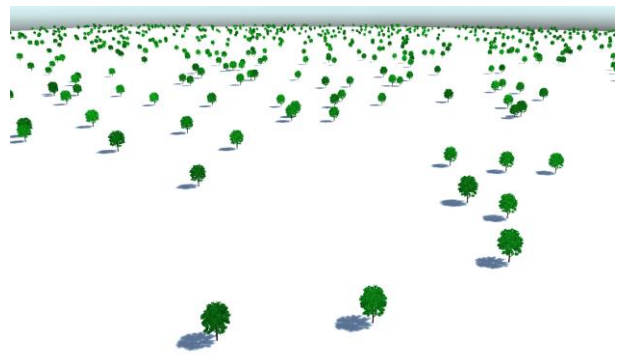
Do badań stworzono podobne do siebie aplikacje wykorzystujące te same modele kuli i podobne skrypty. Aplikacje, które generują modele 3D kuli zaprogramowano tak by liczbę kul można było wpisać w polu tekstowym. W trakcie badań wygenerowano 1000, 5000, 10000 i 15000 kul na które działa grawitacja. Kule podczas testów poruszają się i odbijają się od siebie. Obiekty umieszczane są w zamkniętej platformie tak by miały one ze sobą jak najczęstszy kontakt. Dla każdego obiektu zachowane są zasady dynamiki i istnieje minimalna siła tarcia. Taka aplikacja

pozwała obciążyć komputer na tyle by sprawdzić jak poszczególne silniki radzą sobie z tym zadaniem (Rysunek 1).



Rysunek 1: Aplikacja generująca modele 3D kul na silniku Flax Engine.

Drugi rodzaj aplikacji wykorzystuje model drzewa. Model ten składa się z 44848 wielokątów [11]. W aplikacji stworzono płaski teren, na którym umieszczono w losowych miejscach wspomniany model drzewa w liczbie 250, 500 i 1000. Odległość rysowania cieni ustawiono na zero (brak cienia), połowę długości terenu (odległość średnia) i maksymalną (cały cień). Przykładowa aplikacja została pokazana na Rysunku 2.



Rysunek 2: Aplikacja lasu ze średnim dystansem rysowania cieni i z 1000 drzew na silniku Unity.

Oba badania sprawdzały takie parametry jak: liczba klatek na sekundę, średnie zużycie wątku procesora w procentach, zużycie karty graficznej w procentach i średnie zużycie RAM w MB. By zarejestrować te parametry użyto autorskiego programu napisanego w języku Python. W programie tym użyto biblioteki GPUtil [12] do odczytania wykorzystanie karty graficznej oraz biblioteki psutil [13] do odczytania zużycia wątku procesora i zużycia RAM. Do wyznaczenia liczby klatek na sekundę użyto skryptu napisanego w testowanych aplikacjach. Wyniki synchronizowano czasem uniksowym [14].

Badania zostały przeprowadzone na komputerze o parametrach przedstawionych w Tabeli 2.

Tabela 2: Parametry komputera

Element	Wersja
System	Windows 10
Procesor	AMD Ryzen 5 5600X
Pamięć RAM	32GB DDR3
GPU	GeForce RTX 3070
Rozdzielczość	1920x1080

#### 4. Wyniki

Pierwsze testy wykonano na aplikacji generującej modele 3D kul. Wyniki przedstawiono w tabelach dla różnych obciążeń

Tabela 3: Wyniki pomiarów dla obciążenia 1000 kul

	FPS	CPU	RAM [MB]	GPU
Flax	144,11	3,75%	165,94	43,85%
Unity	143,99	6,61%	129,14	37,45%

Pierwszy test (Tabela 3) okazał się zbyt małym obciążeniem by pokazać znaczącą różnicę w wynikach dla FPS. Pozostałe wyniki pokazują, że Unity zużywa więcej mocy obliczeniowej procesora, ale za to wykorzystuje mniej pamięci RAM i karty graficzne

Na podstawie testu drugiego (Tabela 4), podobnie jak na podstawie testu pierwszego, nie można określić znaczącej różnicy w liczbie FPS dla poszczególnych silników, co nadal uniemożliwia wybranie lepszego silnika pod względem wydajności. Pod względem zużycia podzespołów wynik zostaje taki sam jak w teście pierwszym, czyli silnik Unity zużywa więcej procesora natomiast Flax więcej RAM i karty graficznej.

Tabela 4: Wyniki pomiarów dla obciążenia 5000 kul

	FPS	CPU	RAM [MB]	GPU
Flax	145,60	13,08%	215,34	75,56%
Unity	144,00	24,68%	170,67	37,65%

Tabela 5: Wyniki pomiarów dla obciążenia 10000 kul

	FPS	CPU	RAM [MB]	GPU
Flax	58,96	16,08%	278,76	55%
Unity	107,42	37,44%	229,56	39,43%

Tabela 6: Wyniki pomiarów dla obciążenia 15000 kul

	FPS	CPU	RAM [MB]	GPU
Flax	30,94	17,08%	365,43	40,29%
Unity	55,46	42,93%	294,02	35,72%

Wyniki testu trzeciego i czwartego (Tabela 5 i Tabela 6) ukazują, który silnik radzi sobie lepiej z symulacją fizyki z dużą liczbą obiektów 3D. Unity osiągnął większą liczbę klatek na sekundę w obu testach. Unity zużywa więcej procesora, ale za to mniej pamięci RAM i karty graficznej.

Tabela 7: Brak cienia

250 drzew				
	FPS	CPU	RAM [MB]	GPU
Flax	144,50	3,53%	163,35	66,30%
Unity	143,89	2,35%	134,78	35,79%
500 drzew				
	FPS	CPU	RAM [MB]	GPU
Flax	144,19	4,55%	165,93	88,86%
Unity	143,83	4,08%	139,02	51,33%
1000 drzew				
	FPS	CPU	RAM [MB]	GPU
Flax	111,02	1,69%	165,32	97,35%
Unity	132,77	6,11%	144,68	91,40%

Tabela 8: Średni cień

250 drzew				
	FPS	CPU	RAM [MB]	GPU
Flax	136,23	2,55%	180,44	95,02%
Unity	143,91	3,77%	142,37	66,07%
500 drzew				
	FPS	CPU	RAM [MB]	GPU
Flax	90,34	1,27%	182,30	98,00%
Unity	121,16	6,18%	151,23	97,47%
1000 drzew				
	FPS	CPU	RAM [MB]	GPU
Flax	54,71	0,87%	177,71	98,76%
Unity	62,70	5,58%	174,27	98,25%

Wyniki testów na drugiej aplikacji pokazano w Tabelach 7, 8, 9. Unity i w tych przypadkach osiągnęło większą liczbę klatek na sekundę za wyjątkiem aplikacji, w której drzewa nie rzucały cienia przy liczbie drzew 250 i 500 (Tabela 7). Tam liczba klatek na sekundę osiągnęła w obu silnikach maksymalną wartość. Wyniki zużycia podzespołów pokazują, że Unity zużywa więcej procesora, gdy drzewa rzucają cień, ale zużywa mniej RAM i karty graficznej w pozostałych przypadkach. Silnik Flax zużywa stałą ilość RAM w zależności od dystansu

cieniowania natomiast Unity ma tendencję do zwiększania zużycia RAM o niewielką ilość, gdy zwiększa się liczba drzew.

Tabela 9: Maksymalny cień

250 drzew				
	FPS	CPU	RAM [MB]	GPU
Flax	119,49	1,88%	163,72	97,61%
Unity	139,75	5,76%	163,63	94,56%
500 drzew				
	FPS	CPU	RAM [MB]	GPU
Flax	70,66	1,04%	166,22	98,89%
Unity	87,48	5,57%	169,26	97,17%
1000 drzew				
	FPS	CPU	RAM [MB]	GPU
Flax	39,45	0,73%	162,28	99,00%
Unity	42,27	5,26%	178,67	97,45%

## 5. Wnioski

Testy na aplikacjach stworzonych na silnikach Unity i Flax pozwalają na sformułowanie następujących wniosków:

- Unity osiąga lepsze wyniki w liczbie wyświetlanych klatek na sekundę dla obu rodzajów testów,
- Flax zużywa więcej pamięci RAM i procentowo karty graficznej. Sama pamięć jest wykorzystywana w stałej ilości,
- Unity wykorzystuje więcej procesora a pamięć RAM wykorzystuje zależnie od potrzeb.

Wyniki potwierdzają tezę, którą postawiono w rozdziale drugim. Silnik Unity jest wydajniejszy od silnika Flax we wszystkich przeprowadzonych testach. Silnik ten dobrze radzi sobie z obliczaniem fizyki jak i rysowaniem cieni dla skomplikowanych obiektów jak drzewo wykorzystane w aplikacji.

## Literatura

- [1] Dane o liczbie wydawanych gier w poszczególnych latach, <https://steamdb.info/stats/releases/?tagid=0>, [14.01.2022].
- [2] Informacje o Flax Engine, <https://flaxengine.com/blog/category/releases/>, [12.06.2022].
- [3] Dane dotyczące popularności silników gier na Steam, <https://steamdb.info/tech/>, [14.01.2022].
- [4] Licencja personalna dla Unity, <https://unity3d.com/unity/activation/personal>, [12.06.2022].
- [5] Licencja dla Flax, <https://flaxengine.com/licensing/>, [12.06.2022].
- [6] H. Żukowski, Porównanie wydajności trójwymiarowych gier z użyciem silników CryEngine i Unity, Journal Computer Sciences Institute 13 (2019) 345-348, <https://doi.org/10.35784/jcsi.1330>.
- [7] P. Skop, Porównanie wydajności silników gier na wybranych platformach, Journal Computer Sciences Institute 7 (2017) 116-119.
- [8] A. Barczak, H. Woźniak, Comparative Study on Game Engines, STUDIA INFORMATICA Systems and information technology 1-2 (2019) 5-24, <https://doi.org/10.34739/si.2019.23.01>.
- [9] E. Christopoulou, S. Xinogalos, Overview and Comparative Analysis of Game Engines for Desktop and Mobile Devices, International Journal of Serious Games 4 (2017) 21-36, <https://doi.org/10.17083/ijsg.v4i4.194>.
- [10] A. M. Ciekankowska, A. K. Kiszczak – Gliński, K. Dziedzic. Analiza porównawcza wydajności silników Unity i Unreal Engine w aspekcie tworzenia wirtualnych pokazów modeli pochodzących ze skanowania 3D, Journal Computer Sciences Institute 20 (2021) 247-253, <https://doi.org/10.35784/jcsi.2698>.
- [11] Model drzewa, <https://www.turbosquid.com/pl/3d-models/trees-realistic-3d-1218366>, [12.06.2022].
- [12] Kod źródłowy biblioteki GPUUtil, <https://github.com/anderskm/gputil>, [12.06.2022].
- [13] Kod źródłowy biblioteki psutil, <https://github.com/giampaolo/psutil>, [12.06.2022].
- [14] Opis czasu uniksowego, [https://en.wikipedia.org/wiki/Unix\\_time](https://en.wikipedia.org/wiki/Unix_time), [12.06.2022].