

Badanie wydajności silników gier komputerowych

Gabriel Buczkowski, Kamil Olszewski

*Wydział Informatyki, Uniwersytet Kazimierza Wielkiego
Mikołaja Kopernika 1, 85-074 Bydgoszcz*

Streszczenie: *Celem pracy jest przeprowadzenie badania wydajności silników gier z wykorzystaniem wbudowanych w nie narzędzi. Testy wydajności zostaną przeprowadzone dla różnych konfiguracji sprzętowych. W celu realizacji badania powstanie uproszczona gra 3D z widokiem pierwszoosobowym. Gra zostanie zaimplementowana w różnych silnikach gier: Unity, Unreal Engine, CryEngine, Godot oraz Unigine, co pozwoli zebrać dane do analizy by sprawdzić, który z nich będzie najwydajniejszy.*

Słowa kluczowe: *gra komputerowa, silnik gry, widok z pierwszej osoby, unity, Unreal Engine, Godot, Unigine, CryEngine.*

Testing the performance of computer games

Abstract: *The purpose of this is to conduct performance testing of game engines using their built-in tools. Performance tests will be conducted for various hardware configurations. In order to implement the study, a simplified 3D game with a first-person view will be created. The game will be implemented in various game engines: Unity, Unreal Engine, CryEngine, Godot and Unigine, which will allow us to collect data for analysis to see which one will be the most efficient.*

Keywords: *computer game, game engine, First Person View, Unity, Unreal Engine, Godot, Unigine, CryEngine.*

1. WSTĘP

Silniki gier komputerowych, są platformami, które pozwalają na tworzenie gier komputerowych. Dzięki nim możemy integrować oraz łączyć poszczególne elementy gry m.in.: animacje, interakcja z użytkownikiem oraz wykrywanie kolizji, która występuje między obiektami. Fundamentami silników są komponenty, które zapewniają niezbędną funkcjonalność do tworzenia gier. Komponentami, które posiada każdy silnik gier są:

- Moduł audio – głównym celem modułu jest generowanie efektów dźwiękowych w grze.
- Silnik renderujący – pozwala on konwertować dane wejściowe na liczbę pikseli, które są wyświetlane na ekranie.
- Moduł sztucznej inteligencji – pozwala on na dostarczenie technik, dzięki którym możemy definiować

zasady zachowania z postaciami niekontrolowanymi przez graczy.

- Moduł animacji – dzięki temu komponentowi mamy możliwość animowania np. ruchu, a także zmiany kształtu obiektów wirtualnych.
- Moduł sieciowy – umożliwia granie w gry komputerowe, korzystając z urządzeń podłączonych do Internetu.
- Mechanika gry – składa się z elementów, które rządzą światem wirtualnym, umiejętności postaci kontrolowanych przez graczy, postaciami, które są prowadzone przez sztuczną inteligencję, obiekty w wirtualnym świecie, cele, jak i również zadania graczy.
- Narzędzia programistyczne – optymalizacja oraz zwiększająca efektywność pracy z silnikiem gry. Zapewniają również możliwość dodawania wielu typów elementów do gry: animacji, efektów dźwiękowych, po algorytmy sztucznej inteligencji (rys. 1) [1].



Rys. 1. Przykładowy widok gry uruchomionej na silniku Unreal Engine.

2. METODYKA

Omówimy tutaj silniki gier, które zostały wykorzystane do tworzenia gry, języki programowania, programy graficzne oraz środowiska implementacyjne, które zostały wykorzystane, aby wykonać grę komputerową z widokiem pierwszoosobowym.

Silnik Unreal Engine: Został stworzony w 1998 roku przez firmę Epic Games. Na początku silnik pozwalał jedynie modyfikować oraz dodawać nowe elementy do gry Unreal, na potrzeby których został on stworzony. Dziś Unreal Engine pozwala na implementację gier na wielu platformach m.in: Windows, Linux, Mac OS, iOS, Android, PlayStation, Xbox, HTML, SteamVR, HTC Vive, Oculus Rift. Silnik obsługuje pisanie skryptów w języku C++, oraz jest wyposażony w system Blueprint, który wykorzystuje diagramy graficzne, dzięki któremu to mamy możliwość implementacji logiki gry. Logika, która jest napisana za pomocą diagramów graficznych całkowicie zastępuje potrzebę pisania kodu w języku C++.

Silnik składa się z elementów takich jak:

- Silnik Dźwiękowy: Sound Cue,
- Silnik Fizyki: PhysX,
- Silnik Graficzny: Wspiera DirectX 11/12, OpenGL, JavaScript, Bibliotekę WebGL,
- Szkielet gry: Funkcje, które kontrolują przebieg rozgrywki,
- Efekty postprocesowe,
- Sztuczna inteligencja,
- Moduł sieciowy [1].

Pomysłodawcą oraz twórcą silnika CryEngine była firma Crytek, która w 2002 zaprezentowała silnik jedynie w wersji demo technologicznego stworzonego dla Nvidii. W roku 2004 przekształcił się w pełnoprawny silnik gry, gdzie po raz pierwszy wykorzystano go w grze FarCry. Pierwotnie silnik pozwalał jedynie na tworzenie gier FPS, lecz zmieniło się to w 2016 roku, kiedy zadebiutowała najnowsza wersja silnika, czyli CryEngine V. W obecnym kształcie silnik pozwala na tworzenie gier na platformach takich jak: PC, PlayStation, Xbox, a także Oculus Rift. Kod gry może zostać napisany w jednym z dwóch języków programowania C++ oraz C#. Silnik wyróżnia się wysoką jakością efektów graficznych, jak i również systemem graficznym Flow Graph, który umożliwia sterowanie logiką gry z poziomów diagramów graficznych [1].

Silnik Unity został opracowany przez firmę Unity Technologies. Umożliwia on tworzenie gier 2D oraz 3D. W 2005 roku po raz pierwszy został zaprezentowany na targach Apple Worldwide Developers Conference. We wczesnym stadium silnik obsługiwał platformę MacOS, jednak wraz z kolejnymi wydaniem deweloperzy zwiększali liczbę obsługiwanych platform. Od wersji 2018.1.5, Unity pozwala tworzyć gry na wielu platformach, w tym: PC, Linux, Mac OS, PlayStation, Xbox, Nintendo, Android, iOS, Oculus Rift, GearVR. Unity pozwala na pisanie logiki gry w języku programowania C#. Silnik Unity wyposażony jest w komponenty takie jak modułowy system do budowania obiektów. Ten system pozwala na budowanie zestawu komponentów, gdzie komponenty silnika to zestaw funkcjonalności. Rozwiązanie to pozwala na kombinację funkcji zamiast dziedziczenia, które jest używane jako standardowe podejście do programowania zorientowanego obiektowo. Zaletą tego typu podejścia jest zwiększona elastyczność, która pozwala na sprawne i bezproblemowe dopasowywanie komponentów bez konieczności dziedziczenia podczas każdej zmiany na obiektach [1].

Silnik Godot jest to silnik, który został zaprojektowany przez Juana Linietsky'ego oraz Ariela Manzura w roku 2014. Godot jest silnikiem zorientowanym zarówno na tworzenie gier 2D oraz 3D. Koncentruje się on na dostarczaniu dobrze zaokrąglonego zestawu narzędzi do programowania, w tym wbudowanego edytora kodu, silnika renderowania grafiki, narzędzi do odtwarzania dźwięku, narzędzi do animacji i innych. Z biegiem czasu silnik ogromnie się rozrósł, aby móc objąć jeszcze więcej głównym współtwórców, a jednocześnie przyjmować pomoc od społeczności Godot nad dalszym rozwojem silnika [2].

Silnik Unigine: Silnik zaprojektowany został z myślą o obsłudze wirtualnych światów o bezprecedensowej skali bez ograniczeń takich jak:

- Transport od wybrzeża do wybrzeża.
- Szczegółowe projekty urbanistyczne
- Ogólnokrajowe operacje wojskowe
- Symulatory lotu w skali kontynentu
- Symulacja misji kosmicznej

Dzięki podwójnej precyzji współrzędnych w UNIGINE możliwe jest osiągnięcie rzeczywistej skali wirtualnych scen tak dużych jak Układ Słoneczny. Oferuje 64-bitową podwójną precyzję matematyczną (zamiast zwykłej 32-bitowej precyzji) do definiowania współrzędnych obiektów jakie możemy znaleźć w wirtualnej scenie. Dzięki tej technologii istnieje możliwość tworzenia praktycznych oraz nieograniczonych światów o najwyższym poziomie szczegółowości [3].

Język programowania C++: obiektowy język programowania, który został stworzony przez Bjorna Stroustrupa jako część ewolucji języków z rodziny C. Został on opracowany jako ulepszenie języka C, aby zapewnić programistom wyższy stopień kontroli nad pamięcią, oraz zasobami systemowymi. Dzięki wysokiej elastyczności oraz skalowalności, C++ używany jest do tworzenia szerokiej gamy oprogramowania, aplikacji, przeglądarek, graficznych interfejsów użytkownika (GUI), oraz systemów operacyjnych jak i gier. C++ jest językiem, który może bardzo łatwo działać na różnych systemach operacyjnych lub platformach [4].

Język programowania C#: obiektowy język programowania, który służy do tworzenia sieci i stron internetowych. Język C# określany jest jako język infrastruktury języka wspólnego (CLI). Język ten został opracowany w 1999 roku przed Andersa Hejlsberga wraz z jego zespołem jako uzupełnienie do platformy .NET firmy Microsoft. C# posiada ścisły typ zmiennych danych logicznych m.in. bool, podczas gdy ten typ danych może być zwracany jako liczba całkowita lub wskaźnik, aby uniknąć typowych błędów programowania. C# automatycznie zarządza pamięcią niedostępnych obiektów za pomocą modułu garbage collector, który eliminuje problemy programistów i wycieki pamięci. C# jest językiem bezpiecznym, ponieważ posiada domyślną konwersję, która jest implementowana podczas kompilacji lub środowiska uruchomieniowego.

Język programowania Python: interpretowalny język programowania, który wyklucza potrzebę kompilowania

kodu przed wykonaniem, ponieważ jest językiem wysokiego poziomu jest w stanie wyodrębnić szczegóły z kodu. Język ten skupia swoją uwagę na abstrakcji z możliwością zrozumienia przez większość początkujących programistów. Kod pisany w Pythonie jest stosunkowo krótki oraz w porównaniu z językami skompilowanymi takim jak C i C++ wykonuje program wolniej. Jego prostota w konstrukcji składni sprowadza się do tego, że jest on bardzo popularny wśród programistów, głównie zajmującymi się zagadnieniem uczenia maszynowego w aplikacji LCNC (Low-code, no-code). Python posiada prostą składnię oraz posiada dużą społeczność, która aktywnie przyczynia się do rosnącego wyboru modułów oprogramowania i bibliotek. Python został napisany przez Guido van Rossum pod koniec 1980 roku. Jest on obecnie zarządzany przez Python Software Foundation.

Blender to darmowy program na licencji open source, który jest utrzymany przez Blender Foundation. Posiada nieograniczone możliwości, takie jak tworzenie animacji komputerowej czy tworzenie modeli 3D. Posiada on również możliwość renderowania, komponowania, śledzenia ruchu, a także daje możliwość tworzenia w nim gier komputerowych. Użytkownicy, którzy są już bardziej zaawansowani, wykorzystują API Blendera do pisania skryptów Pythona, aby można było dostosować aplikację oraz napisać specjalistyczne narzędzia. Jest to narzędzie wieloplatformowe, które może działać na komputerach Linux, Windows, a także Macintosh. Interfejs wykorzystywany jest w OpenGL, w celu potwierdzenia konkretnej zgodności [5,6].

Adobe Photoshop to oprogramowanie opracowane przez firmę Adobe, które służy do tworzenia obrazów, projektowania graficznego oraz edycji zdjęć. Zapewnia edycję obrazów opartych na grafice rastrowej i wektorowej. Jest to również program do manipulacji obrazami. Wykorzystuje on system edycji, który jest oparty na warstwach. Photoshop został stworzony w 1998 roku przez Thomasa Knolla oraz Johna Knolla, a oficjalną licencję dystrybucyjną programu zakupiła firma Adobe Systems. Photoshop stał się integralną częścią wielu różnych branż, w tym publikacji, projektowania stron internetowych, medycyny, filmu, reklamy, inżynierii, a także architektury [7-9].

Visual Studio to zintegrowane środowisko programistyczne (IDE), które jest bogate w funkcje obsługujące wiele aspektów tworzenia oprogramowania. Visual Studio służy do edytowania, debugowania oraz kompilowania kodu, a następnie umożliwia publikowanie stworzonych aplikacji. Zawiera również narzędzia do uzupełnienia kodu, projektantów graficznych i wiele innych

funkcji, które usprawniają proces tworzenia oprogramowania. Program Visual Studio istnieje od 20 lat. Jego pierwsza wersja było Visual Studio 97. Zawiera ono edytor kodu obsługujący technologię IntelliSense (składnik uzupełnienia kodu), a także refaktoryzację kodu. Inne wbudowane narzędzia obejmują profiler kodu, możliwość tworzenia GUI, projektowania stron internetowych oraz projektowanie schematu baz danych [10,11].

3. TESTOWANIE SILNIKÓW GIER W OPARCIU O STWORZONĄ GRĘ KOMPUTEROWĄ

Jeżeli chcemy odpowiednio sprawdzić, czy dany silnik gier nadaje się do tworzenia gry, którą chcemy zrobić, należy w pierwszej kolejności stworzyć tzw. demo projektu, aby stwierdzić czy dany silnik pozwala odpowiednio zoptymalizować grę oraz, że nie tylko będzie działać płynnie, lecz będziemy mieć możliwość wyciągnięcia jak najlepszych efektów graficznych, aby dany gracz miał jak najlepsze wrażenia grając w naszą grę komputerową. Dlatego każdy silnik został sprawdzony na 10 platformach, dzięki czemu będzie możliwość sprawdzenia, który silnik gier jest najbardziej optymalny.

Wykorzystano następujące platformy testujące:

Test 1: Laptop

Procesor: i7-11370H 3.30 GHZ
Pamięć Ram: 24 GB RAM DDR4
Karta Graficzna: RTX 3070

Test 2: Laptop

Procesor: i5-7400 3GHZ
Pamięć Ram: 8GB RAM DDR4
Karta Graficzna: GTX 1050

Test 3: Komputer

Procesor: i7-9750h 2.60 GHZ
Pamięć Ram: 16 GB RAM DDR4
Karta Graficzna: GTX 1660 TI

Test 4: Komputer

Procesor: AMD FX-8320E 3.20 GHZ
Pamięć Ram: 10 GB RAM DDR4
Karta Graficzna: GTX 750 TI

Test 5: Komputer

Procesor: i7 8550U, 1,8 GHZ
Pamięć Ram: 16 GB RAM DDR3
Karta Graficzna: MX150

Test 6: Komputer

Procesor: i7 9700K 3.60GHZ
Pamięć Ram: 16 GB RAM DDR4
Karta Graficzna: RTX 2070

Test 7: Komputer

Procesor: AMD Ryzen 5 2600 3.70 GHZ
Pamięć Ram: 16GB RAM DDR4
Karta Graficzna: GTX 1060

Test 8: Komputer

Procesor: i5-7200U. 3.10 GHZ
Pamięć Ram: 8GB RAM DDR3
Karta Graficzna: GTX 940 MX

Test 9: Komputer

Procesor: i9-10900KF 3.70 GHZ
Pamięć Ram: 32 GB RAM DDR4
Karta Graficzna: RTX 3070

Test 10: Laptop

Procesor: i5-10300H 2.50 GHZ
Pamięć Ram: 8GB RAM DDR4
Karta Graficzna: GTX 1650 TI.

Najlepsze testy na danym silniku gry zostały omówione poniżej.

Unreal Engine: test 7 dający wyniki:

Procesor (CPU): Obciążenie na poziomie 3.65 GHZ
Ram: 5.4/15.9 GB
Karta Graficzna (CPU): 3.9 GB/6.0 GB
Ilość Klatek na sek. (FPS): 51 FPS

CryEngine: test 2 dający wyniki:

Procesor (CPU): Obciążenie na poziomie 3.33 GHZ
Ram: 5.4/14.2 GB
Karta Graficzna (CPU): 2.9 GB/4.0 GB
Ilość Klatek na sek. (FPS): 44 FPS

Unity: test 5 dający wyniki:

Procesor (CPU): Obciążenie na poziomie 1.99 GHZ
Ram: 2.8/8.0 GB
Karta Graficzna (CPU): 0.75 GB/1.5 GB
Ilość Klatek na sek. (FPS): 47 FPS

Godot: test 9 dający wyniki:

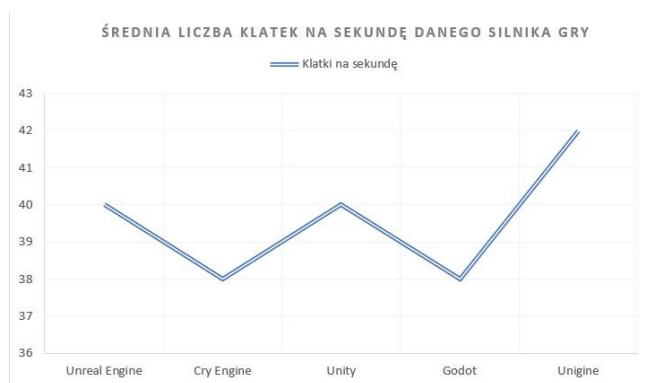
Procesor (CPU): Obciążenie na poziomie 4.68 GHZ
Ram: 5.8/31.9 GB

Karta Graficzna (CPU): 2.7 GB/8.0 GB
Ilość Klatek na sek. (FPS): 43 FPS

Unigine: test 10 dający wyniki:
Procesor (CPU): Obciążenie na poziomie 4.12 GHZ
Ram: 5.4/7.9 GB
Karta Graficzna (CPU): 1.3 GB/4.0 GB
Ilość Klatek na sek. (FPS): 50 FPS

4. PODSUMOWANIE TESTÓW

Każdy z powyższych testów został wykonany na innej platformie pod względem parametrów komputera oraz na innym silniku gry. Sprawdzane zostały parametry takie jak: Procesor (CPU), RAM, Karta Graficzna (GPU), oraz ilość klatek na sekundę jaka była wyświetlana podczas rozgrywki. W niektórych sytuacjach mogliśmy zaobserwować błędy, które uniemożliwiały nam uruchomienie gry na danym silniku gry. Najczęstsze problemy jakie zostały napotkane to odpowiednie ustawienie oświetlenia mapy, aby wszystko co znajduje się na mapie było widoczne podczas wykonywania testów, nakładanie tekstur, blokowanie programu wykonywanego przez program antywirusowy. Jeżeli chodzi o ranking silników pod względem obciążeń podzespołów komputerów, które zostały wykorzystane podczas testowania gry to silnikiem, który najbardziej obciążał sprzęt podczas testów był silnik CryEngine, następnym silnikiem był Unigine, kolejnym był Unreal Engine (rys. 2.).



Rys. 2. Średnią liczbę klatek na sekundę dla poszczególnych silników gier.

5. WNIOSKI

Cel projektu, który był do zrealizowania, został ukończony. Wykorzystano do tego silniki gier, dzięki którym mogliśmy zaprojektować identyczną grę na różnych silnikach gier, dzięki czemu mogliśmy zobaczyć, gdzie dana gra działa najlepiej na danym silniku gier. Jeżeli chodzi natomiast o problemy, które wynikały podczas projektowania gry, możemy wywnioskować, że jest ich bardzo dużo z uwagi na fakt dodanej ilości elementów, jakie wystąpiły podczas tworzenia danej gry komputerowej.

Literatura

- [1] Woźniak H, Barczak A. Badanie porównawcze silników gier. Comparative Study on Game Engines. 2019; p. 5--24.
- [2] Schardon L. What is Godot? The Free Engine for Making 2D & 3D Games. [Online].; 2023 [cited] 2023 Luty 16. Available from: <http://gamedevacademy.org/what-is-godot/>.
- [3] Unigine. UNBOUNDED WORLDS. [Online].; 2015 [cited] 2023 Luty 16. Available from: <https://web.archive.org/web/20190419073000/https://unigine.com/en/products/engine/unbounded-world>.
- [4] Buttice C, Rabelo J. Techopedia. [Online].; 2021 [cited] 2023 Luty 16. Available from: <https://www.techopedia.com/definition/26184/c-plus-plus-programming-language>.
- [5] Blain JM. An Introduction to Blender 3D A Book For Beginners: Blender; 2011.
- [6] Blender. The Freedom to Create. [Online].; 2023 [cited] 2023 Luty 16. Available from: <https://www.blender.org/about/>.
- [7] Hanna KT. TechTarget. [Online].; 2023 [cited] 2023 Luty 16. Available from: <https://www.techtarget.com/whatis/definition/Photoshop>.
- [8] Walker A. Guru99. [Online].; 2023 [cited] 2023 Luty 16. Available from: <https://www.guru99.com/introduction-to-photoshop-cc.html>.
- [9] Encyclopaedia Britannica. Britannica. [Online].; 2023 [cited] 2023 Luty 16. Available from: <https://www.britannica.com/technology/Adobe-Photoshop>.
- [10] Microsoft. Microsoft. [Online].; 2022 [cited] 2023 Luty 16. Available from: <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>.
- [11] Incredibuild. Incredibuild. [Online].; 2023 [cited] 2023 Luty 16. Available from: <https://www.incredibuild.com/integrations/visual-studio>.

