

АНАЛІЗ API DIRECTX 12

Вінницький національний технічний університет

Анотація

У статті проаналізовано особливості нового пакету API корпорації Microsoft під назвою «DirectX 12». Метою роботи є аналіз DirectX 12.

Ключові слова: DirectX 12, драйвер, шейдери, програміст, продуктивність роботи, API DX12, GPU.

Вступ

У той час як продуктивність GPU стрімко зростає, потенціал розвитку одноядерних CPU був обмежений енергоспоживанням, що зображено на рис.1[1]. Багатоядерні CPU забезпечили підвищення швидкодії обробки інформації, проте їх пікова продуктивність все ще не досягала таких показників як у GPU

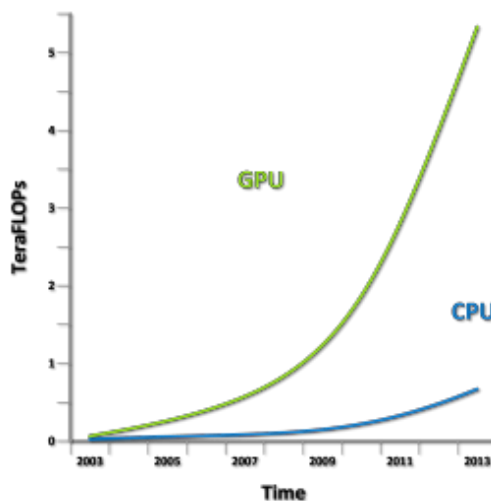


Рис. 1. Розвиток продуктивності CPU та GPU

. Одночасно додатки стали використовувати паралелізм на рівні завдань, використовуючи складні планувальники для масштабування продуктивності при збільшенні кількості ядер центрального процесора. Це викликало необхідність у створенні API для контролю розподілу роботи між ядрами. DirectX 12 може слугувати прикладом ефективного використання API. Очевидно, що це є ефективним ресурсом для сучасної ігрової індустрії. 20 березня 2014 року відбувся анонс DirectX 12. Розробники внесли ряд змін у DirectX 3D, що дало можливість пришвидшити та збільшити ефективність багатьох графічних операцій. Відповідно, витоки DX12 можна прослідкувати у тенденціях розвитку технологій. Нові зміни дозволяють створювати більш детальні сцени та досягти повного використання можливостей сучасних GPU.

Результати дослідження

Співпраця корпорацій NVIDIA та Microsoft по створенню нової технології DX 12 була основана на потребі зменшення витрат на управління ресурсами GPU[1-5]. Було прогнозовано, що розробка нового низькорівневого API буде закінчено у березні 2014 року, проте робочу версію було представлено 29 липня 2015 року, після виходу Windows 10, що була єдиною ОС, що підтримувала DirectX 12. Сьогодні провідні компанії NVIDIA та AMD забезпечують підтримку DirectX 12 на усіх ОС Microsoft. 70% усіх комп'ютерів (GPU на базі архітектур Fermi, Kepler і Maxwell), що підтримували DX11, також можуть оперувати API DX12. У 2018 році на конференції GDC (Microsoft) представили оновлення для DirectX 12 - Microsoft DXR, яке дозволяє API виконувати трасування променів в реальному часі.

Драйвери і програмне забезпечення ОС раніше керували пам'яттю й іншими ресурсами комп'ютера. Однак це виявилось вичерпним сьогодні. Низька продуктивність пов'язана в першу чергу з тим, що алгоритми ОС не взаємодіяли з ПЗ додатка. DirectX 12 у свою чергу безпосередньо оперує ресурсами комп'ютера та виконує синхронізацією апаратного забезпечення. В результаті розробники ігор можуть налагодити ефективну роботу GPU щодо інших компонентів.

Основним завданням нового покоління DirectX – оптимізація. Продукт Microsoft може працювати з оновленням програмного забезпечення до Windows 10 (і DirectX 12 відповідно). Мультимедійна (тобто ігрова) продуктивність комп'ютера в теорії може збільшитися майже у два рази. Досягти такого збільшення продуктивності без заміни апаратних засобів можливо за рахунок повноцінного використання ресурсів. Звичайно заміна одноядерного на дво- або чотирьох ядерний процесор, безсумнівно, сприяє зростанню продуктивності, але не так, як цього очікують користувачі. Завдання нового DirectX 12 полягає в тому, щоб дозволити високопродуктивному ПЗ по максимуму використовувати всі доступні ресурси ядер процесора. Результати такої оптимізації в тесті 3D Mark показують приріст продуктивності на 10-15% порівняно DirectX 11 і DirectX 12, як зображено на рис.2 [6]

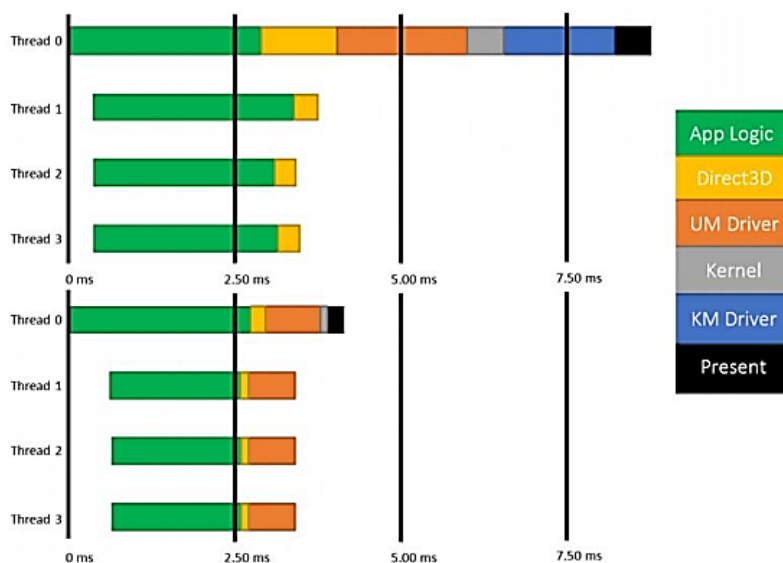


Рис. 2. Тестування на 3DMark

Нова технологія являє собою частину Windows SDK в Windows 10. У якості IDE використовується Visual Studio, мова програмування – C++. DirectX 12 відрізняється опрацюванням управління станів. Нововведенням є відстеження та управління ресурсами в пам'яті. До того ж було добавлено об'єкти конвеєра стану (PSO), які служать ще більшими об'єктами станів разом з шейдерами. Раніше операційна система була основою для прогнозу та відстеження використання ресурсів, що обмежувало можливість створення додатків при широкомасштабному використанні.

У DirectX 12 саме програміст (а не система чи драйвер) відповідає за продуктивність роботи[2]:

- Програма DX12 не веде відстеження прив'язки ресурсу.
- Архітектура оптимізована спеціально під високоякісний та продуктивний код.
- Драйвер нового покоління не опрацьовує самостійно синхронізацію доступу центрального процесора та відео карти.

Єдиним механізмом для цього виступає повне забезпечення контролю над роботою GPU додатком, що у свою чергу дає змогу не виконувати API постійного обрахунку, тобто автоматичного (ресурсами відео карти), а залишити лише зазначені команди опрацьованої програми. У DirectX 12 immediate context був виключений і концепція deferred стала основною, що значить, що програміст повинен заповнити deferred самостійно й у необхідний момент відправити його на виконання.

DirectX 12 використовує технологію fence, яка інкапсульована о ID3D12Fence[3] інтерфейсом. Fence - це ціле число, яке представляє виконану роботу GPU на поточний момент.

Нововведеннями та покращеннями є технології DirectX 12[3]:

- Якісна візуалізація тексту з попереднім звуковим супроводом.
- Профілювання процесора та GPU у режимі реального часу.
- Змінні, керовані користувачем.
- Пряма підтримка взаємодії ігрового контролера, миші та клавіатури.
- Зрозуміла інтуїтивно обгортка DirectXMath.
- Камера перспективи, яка підтримує традиційні та обернені Z матриці.
- Асинхронне завантаження текстур DDS та декомпресія ZLib.
- Велика бібліотека шейдерів.
- Спрощене вбудовування шейдерів за допомогою системи компіляції до виклику.
- Безпечна для потоків система контекстних команд GPU (WIP).
- Прості у використанні динамічні постійні буфери та таблиці дескрипторів.
- Повноцінна підтримка стандартів 4K відео.
- DX12 не знижує навантаження на процесор, а навпаки – збільшує, тільки розподіляється до п'ядкер/потоків процесора, а не лягає на 1 ядро.
- DirectX Raytracing дозволяє моделювати реальну поведінку світла в грі, створюючи неймовірно реалістичні графічні ефекти, такі як глобальне освітлення, відображення й тіні.

Висновки

Отже, нове рішення запропонованого DirectX 12 може бути гарною заміною вже вичерпаного пакету старого покоління. Окрім цього значні покращення у роботі та перспективи для розробників відео ігор забезпечать постійну та надійну підтримку зі сторони корпорації Microsoft Windows. Дана технологія забезпечить приріст потужності як дискретної, так і інтегрованої відеокарти. Основне завдання DirectX 12 направлене на значне покращення якості графіки за рахунок оптимізації. Ефективність зростає близько на 10-15%, поліпшується використання CPU, а також забезпечується більш ефективно розпаралелювання роботи між потоками.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. DirectX 12: новый этап в развитии игр [Електронний ресурс] // nvidia. – 2014. – Режим доступу до ресурсу: <https://www.nvidia.com.ua/object/blog-nvidia-directx-12-ru.html>.

2. Irbis. DirectX 12 Ultimate: универсальный API для карт AMD Navi 2X, Nvidia GeForce RTX и консоли Xbox Series X [Электронный ресурс] / Irbis // overclockers. – 2020. – Режим доступа до ресурсу: <https://www.overclockers.ua/news/hardware/2020-03-20/126492/>.

3. DirectX 12 — от Леонардо да Винчи к современному искусству [Электронный ресурс] // habr. – 2020. – Режим доступа до ресурсу: <https://habr.com/ru/post/489282/>.

4. DirectX-Graphics-Samples [Электронный ресурс] // GitHub. – 2019. – Режим доступа до ресурсу: <https://github.com/Microsoft/DirectX-Graphics-Samples>.

5. Что такое DirectX 12 [Электронный ресурс] – Режим доступа до ресурсу: <https://windowstips.ru/chtotakoe-directx-12-i-kakie-preimushhestva-on-neset>.

6. DirectX 12 [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/company/microsoft/blog/216579/>

Марущак Артем Володимирович — студент першого курсу групи ЗПІ-196, ФІТКІ, Вінницький національний технічний університет, Вінниця, e-mail: maryskhak@gmail.com

Шмалюх Владислав Анатолійович — студент першого курсу групи ЗПІ-196, ФІТКІ, Вінницький національний технічний університет, Вінниця, e-mail: zskat02@gmail.com

Науковий керівник: **Романюк Олександр Никифорович** — д. т.н., професор, завідувач кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: rom8591@gmail.com