УДК 004.92

Є. К. Завальнюк
О. Н. Романюк
О. В. Романюк

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ШЕЙДЕРНИХ ПРОГРАМ ІЗ ВИКОРИСТАННЯМ OPENGL 4.6

Вінницький національний технічний університет;

Анотація
У роботі проаналізовано нові специфікації прикладного програмного інтерфейсу OpenGL 4.6.
Ключові слова: OpenGL, DirectX, API, рендеринг, шейдер.

Abstract
In the work, the main new features of specification of application programming interface OpenGL 4.6 are analyzed.
Keywords: OpenGL, DirectX, API, rendering, shader.

Вступ

Для спрощення розробки шейдерних програм і програмних продуктів доцільно проаналізувати стандартні підходи і прикладні програмні інтерфейси (API). До появи API кожен розробник розробляв власні драйвери [1] для управління графічним апаратним забезпеченням. Окрім того, використовувались невпорядковані набори інтерфейсів [1], що збільшувало час розробки програм. Поява API для формування двовимірних і тривимірних зображень, серед яких OpenGL і DirectX, дозволила суттєво підвищити ефективність розробки графічних додатків.

Метою роботи є аналіз особливостей нововведень API OpenGL версії 4.6.

Аналіз особливостей OpenGL 4.6

Основними прикладними програмними інтерфейсами для розробки графічних додатків є OpenGL і DirectX. DirectX [2] – пропрієтарний набір API від Microsoft, орієнтований на розробку ігор для Windows і Xbox. Характеризується описом широкого функціоналу, що додатково включає роботу зі звуком, відео, мережевими технологіями.

OpenGL (Open Graphics Library) [3, 4] є специфікацією API для розробки систем рендерингу, що оновлюється консорціумом Khronos Group. API характеризується гнучкістю та кросплатформеністю.

Перша версія специфікації OpenGL була випущена у 1992 р. Найновішу версію OpenGL 4.6 [5] було випущено влітку 2017 р.

Одним із основних нововведень є підтримка SPIR–V (Standart Portable Intermediate Representation) шейдерів [5]. Мова SPIR–V використовується для проміжного подання шейдерного коду (наприклад, коду на мові GLSL), що дозволяє його використання у програмах на основі API OpenGL, Vulkan, OpenCL. Окрім комп’ютерної графіки, мова SPIR–V використовується для паралельних обчислень. Стандартизацію підтримки SPIR–V для OpenGL реалізовано у розширеннях GL_ARB_gl_spirv і GL_ARB_spirv_extensions.

Додатково зменшено навантаження на CPU, що пов’язане з пакетним рендерингом геометричних примітивів (розширення GL_ARB_indirect_parameters і GL_ARB_shader_draw_parameters). Розширення GL_ARB_indirect_parameters [6] полягає у введені концепції буфера параметрів, де зберігаються параметри непрямих команд візуалізації. Непряме виконання графічних команд полягає у збереженні їх з параметрами у буфері та подальшому виконанні за один виклик API. Оскільки число виконаних даним способом команд може бути невідоме, використовуються запити зворотного зв'язку перетворень щодо кількості генерованих об'єктів або буфери відображень на основі атомарних лічильників OpenGL. У результаті, сповільнюється формування зображень. Тому застосовуються запропоновані спеціальні буфери параметрів. Розширення

GL_ARB_shader_draw_parameters [7] включає введення параметрів для вершинного шейдера gl_BaseVertexARB і gl_BaseInstanceARB. Зазначені параметри підвищують ефективність доступу до екземплярів полігональних моделей спільного буфера зі зміщеними індексами. Також введено параметр gl_DrawID для індексації команд під час єдиного багатокомандного виклику.

Додано можливість використання анізотропної фільтрації (GL_ARB_texture_filter_anisotropic) [8, 9], що застосовується для підвищення якості накладання текстур на поверхні, які спостерігаються під похилим кутом. Стандартні підходи OpenGL до накладання текстур базуються на припущенні, що проекція пікселя на текстурний простір є однорідною (ізотропною) за формою. Припущення виконується лише у ситуації, коли поверхня паралельна екрану. Це призводить до наявності розмитих і нечітких ділянок поверхні об'єкта. Тому використовується анізотропна фільтрація, що враховує коректну проекцію пікселя на простір текстури. У розширенні реалізовано лише загальний підхід до анізотропної фільтрації.

Забезпечено можливість зменшення рівня навантаження на драйвери шляхом вимкнення генерації помилок (GL_KHR_no_error) [5]. Рекомендується застосовувати даний режим лише після того, як перевірено відсутність помилок у коді. Інакше, при виникненні помилки можливе припинення роботи програми.

Усунено артефакт витоків світла при візуалізації тіней (GL_ARB_polygon_offset_clamp) [10]. Для цього обмежується можливе значення зсуву полігона. Зсув полігона використовується у випадках, коли полігони мають схожі z–глибини й перекривають один одного. Великі значення зсувів можуть спричинити розриви у візуалізованих тінях. Тому для усунення артефакту величину зсуву було обмежено.

Реалізовано можливість створення запитів щодо статистики виконання процесів графічного конвеєра (GL_ARB_pipeline_statistics_query) [11] та запитів для виявлення переповнення буферів зворотного зв'язку перетворень (GL_ARB_transform_feedback_overflow_query). Розширення GL_ARB_pipeline_statistics_query надає можливість отримання інформації про кількість викликів геометричного, вершинного, фрагментного, теселяційного, обчислювального шейдерів, кількість вершин і примітивів для обробки, кількість створених примітивів геометричним шейдером, кількість оброблених ділянок шейдером контролю теселяції, вхідну та вихідну кількість примітивів на етапі відсікання.

Для підвищення продуктивності формування зображень додано нові вбудовані функції шейдерів (GL_ARB_shader_atomic_counter_ops і GL_ARB_shader_group_vote).

Розширення GL_ARB_shader_atomic_counter_ops [12] забезпечує збільшення кількості операцій із атомарними лічильниками (атомарних операцій над буферними об'єктами). До введених операцій належать додання, віднімання, знаходження максимального й мінімального значень, побітові операції, порівняння, обмін. GL_ARB_shader_group_vote [13] забезпечує більш ефективне виконання коду шейдера на процесорах типу SIMD (Single Instruction/Multiple Data – «Одна інструкція/багато даних») [14]. Забезпечено вилучення ситуацій, коли потоки шейдера розходяться відносно логічних умов, внаслідок чого частина потоків є неактивною та чекає виконання. Якщо для усіх потоків умова виконується, обирається відповідна логічна гілка виконання. Інакше обирається альтернативна гілка.

Зазначені розширення додано у ядерну специфікацію OpenGL. У вигляді додаткових розширень [5] реалізовано багатопотокову шейдерну компіляцію (GL_KHR_parallel_shader_compile), вимкнення створення контекстів помилок при роботі з WGL і GLX API (API для взаємодії OpenGL із Windows та X Window відповідно) (WGL_ARB_create_context_no_error і GXL_ARB_create_context_no_error).

Також введено розширення для управління об'єктами пам'яті [5], що забезпечує сумісність OpenGL з Vulkan, DirectX (GL_EXT_memory_object, GL_EXT_semaphore, GL_EXT_win32_keyed_mutex).

Висновки

Нові функції версії OpenGL 4.6 забезпечують більш ефективне виконання шейдерного коду та використання ресурсів CPU, драйверів, підвищення якості накладання текстур на поверхні об'єктів і реалістичності формування тіней.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The History of OpenGL [Online]. – URL: https://www.codingunit.com/the-history-of-opengl. (Accessed on: Feb. 17, 2024). – Title from the screen.

2. Аналіз особливостей DirectX12 / Є. К. Завальнюк [та ін.] // The 9th International scientific and practical conference “Modern research in world science”, [Львів], 28 – 30 лист. 2022 р. / SPC “Sci-conf.com.ua”. – Львів, 2022. – С. 484 – 486.

3. Романюк О. Н. Комп’ютерна графіка: навчальний посібник / О. Н. Романюк, О. В. Романюк, Р. Ю. Чехместрук. – Вінниця: Вінницький національний технічний університет, 2022. – 140 с.

4. Аналіз крос-платформового програмного інтерфейсу OpenGL і його нововведень / О. Н. Романюк [та ін.] // Інформаційні технології і автоматизація – 2021, [Одеса], 21 – 22 жовт. 2021 р. / Одеська національна академія харчових технологій. – Одеса, 2021. – С. 255 – 260.

5. OpenGL Overview [Online]. – URL: https://www.khronos.org/api/opengl. (Accessed on: Feb. 17, 2024). – Title from the screen.

6. ARB_indirect_parameters [Online]. – URL: https://registry.khronos.org/OpenGL/extensions/ARB/ARB_indirect_parameters.txt. (Accessed on: Feb. 17, 2024). – Title from the screen.

7. ARB_shader_draw_parameters [Online]. – URL: https://registry.khronos.org/OpenGL/extensions/ARB/ARB_shader_draw_parameters.txt. (Accessed on: Feb. 17, 2024). – Title from the screen.

8. Романюк О. Н. Аналіз методів анізотропної фільтрації текстур / О. Н. Романюк, С. О. Романюк, М. П. Піддубецька // Вимірювальна та обчислювальна техніка в технологічних процесах. –2013. – №2. – С. 123 – 128.

9. ARB_texture_filter_anisotropic [Online]. – URL: https://registry.khronos.org/OpenGL/extensions/ARB/ARB_texture_filter_anisotropic.txt. (Accessed on: Feb. 17, 2024). – Title from the screen.

10. ARB_polygon_offset_clamp [Online]. – URL: https://registry.khronos.org/OpenGL/extensions/ARB/ARB_polygon_offset_clamp.txt. (Accessed on: Feb. 17, 2024). – Title from the screen.

11. ARB_pipeline_statistics_query [Online]. – URL: https://registry.khronos.org/OpenGL/extensions/ARB/ARB_pipeline_statistics_query.txt. (Accessed on: Feb. 17, 2024). – Title from the screen.

12. ARB_shader_atomic_counter_ops [Online]. – URL: https://registry.khronos.org/OpenGL/extensions/ARB/ARB_shader_atomic_counter_ops.txt. (Accessed on: Feb. 17, 2024). – Title from the screen.

13. ARB_shader_group_vote [Online]. – URL: https://registry.khronos.org/OpenGL/extensions/ARB/ARB_shader_group_vote.txt. (Accessed on: Feb. 17, 2024). – Title from the screen.

14. Завальнюк Є. К. Реалізація паралелізму потоків команд і даних графічних процесорів / Є. К. Завальнюк, О. Н. Романюк // Інноваційні дослідження та перспективи розвитку науки і техніки у XXI столітті, [Рівне], 19 жовт. 2023 р. / Приватний вищий навчальний заклад «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем’янчука. – Рівне, 2023. – С. 156 – 158.

Завальнюк Євген Костятинович — аспірант кафедри програмного забезпечення, факультет інформаційних технологій та комп’ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: qq9272627@gmail.com.

Романюк Олександр Никифорович — д-р техн. наук, професор, завідувач кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця

Zavalniuk Yevhen K. — postgraduate student at Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia, email : qq9272627@gmail.com

Romanyuk Oleksandr N. — Dr. Sc., Professor, Head of the Chair of Software Engineering, Vinnytsia National Technical University, Vinnytsia