

# ВИКОРИСТАННЯ МОЖЛИВОСТЕЙ БІБЛІОТЕКИ OPENGL ПРИ ФОРМУВАННІ ГРАФІЧНИХ ЗОБРАЖЕНЬ

Вінницький національний технічний університет

## *Анотація*

*Наведено короткі відомості про використання графічного процесору за допомогою OpenGL та його можливості.*

**Ключові слова:** комп'ютерна графіка, OpenGL.

## *Abstract*

*Brief information of GPU usage with OpenGL and its capabilities.*

**Keywords:** computer graphics, OpenGL.

## **Вступ**

У сучасному цифровому віці графіка відіграє ключову роль у візуальному сприйнятті інформації та спілкуванні. Графічне програмування, як важлива галузь інформатики, стоїть за створенням неймовірно реалістичних візуальних образів, від ігор і анімацій до віртуальних середовищ та дизайну інтерфейсів [1].

Комп'ютерна графіка – це галузь науки та технології, що займається створенням, обробкою та відображенням графічних зображень і відео за допомогою комп'ютерів. Ця область включає в себе різні аспекти, такі як створення тривимірних та двовимірних зображень, анімація, комп'ютерна візуалізація, обробка зображень, віртуальна реальність та інші.

При роботі з комп'ютерною графікою виникають задачі, які вимагають великої кількості обчислень. Наприклад, визначення кольору для кожного пікселя на екрані може становити значний обсяг роботи. Навіть при розширенні екрану 800x600 необхідно обробити 480 тисяч пікселів. Крім того, для плавної картини ці обчислення потрібно робити не менше ніж 24 рази в секунду. Сучасні центральні процесори не здатні виконувати таку кількість операцій в заданий час, тому в нагоді стане графічний процесор (GPU).

Графічні процесори призначені саме для подібних завдань. Вони мають багато ядер, що дозволяє їм обробляти одночасно велику кількість даних. Це забезпечує значно більшу швидкість обчислень порівняно з центральним процесором для графічних завдань.

Для того щоб застосовувати графічний процесор в програмуванні потрібно скористатись одним з прикладних програмних інтерфейсів (API), які надають драйвери встановленої відеокарти. В сучасному світі є API, які вважаються стандартом для будь-якої популярної компанії виробництва відеокарт. Тому незалежно від марки або версії відеокарти, програми розроблені на таких інтерфейсах будуть працювати й з іншими відеокартами. Один із таких стандартизованих API є OpenGL. Бібліотека OpenGL налічує близько 350 основних функцій, які забезпечують формування складних 3D зображень із використанням 10 видів графічних примітивів [2].

## **Можливості бібліотеки OpenGL**

Можливості бібліотеки OpenGL базуються на гнучкості його графічного конвеєру.

Графічний конвеєр – це послідовність етапів обробки для перетворення даних у остаточне відтворене зображення [3].

Графічний конвеєр OpenGL зображено на рисунку 1. Секції, позначені синім кольором, вказують на можливість користувача створити власну реалізацію цих етапів. У випадку з OpenGL такими етапами є вершинний, геометричний та піксельний шейдери [4].

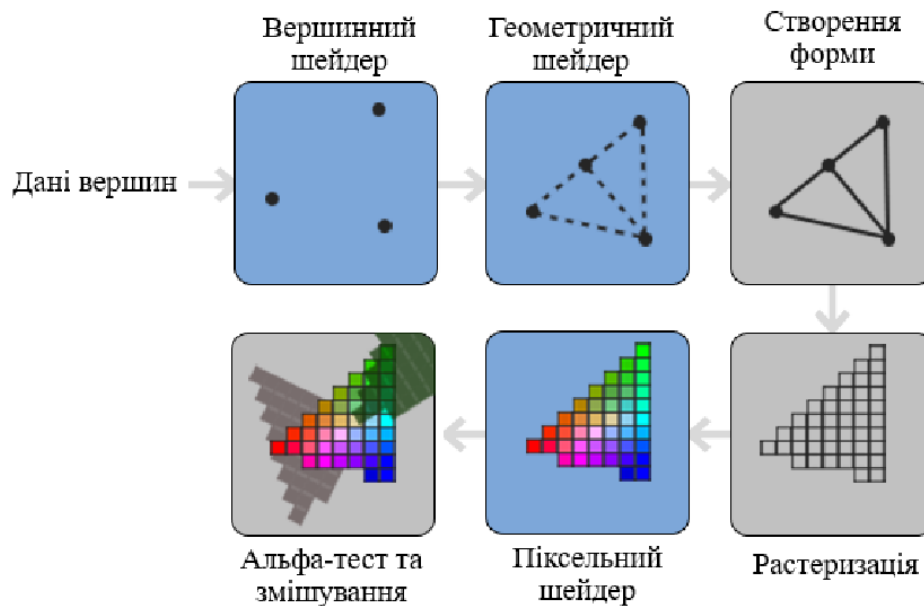


Рисунок 1 – Графічний конвеєр OpenGL

Опис графічного конвеєру OpenGL:

1. **Дані вершин.** Вхідними даними є дані вершин. Зазвичай це тривимірні координати вершин трикутників, з яких формується фінальне зображення, але OpenGL дозволяє користувачу самому вирішувати які дані передавати та яким чином їх обробляти в подальших етапах.

2. **Вершинний шейдер.** Перший етап конвеєру – вершинний шейдер, який приймає як вхідну інформацію одну вершину. Основна задача вершинного шейдера це перетворення 3D-координати в іншу 3D-координату. Найчастіше це виконується для того, що б правильно спроектувати тривимірний простір на двовимірний екран.

3. **Геометричний шейдер.** Геометричний шейдер приймає як вхідні дані колекцію вершин, які утворюють примітив і генерує нові вершини, які формують нові примітиви. Цей етап є опціональним і не використовується так часто як вершинний або піксельний шейдер, хоча може бути не менш корисним інструментом.

4. **Створення форми.** Етап створення форми (або етап формування примітивів) приймає оброблені до нього вершини та збирає в заданий примітив. Користувач повинен задати тип примітива, який буде використано при формуванні. OpenGL дає можливість обрати такі примітиви як точка, лінія або трикутник. За допомогою цих трьох базових примітивів можна створити будь-яку форму, яку побажає користувач.

5. **Растреризація.** Вихідні дані стадії створення форми передаються на стадію растреризації, де отримані примітиви формують пікселі кінцевого зображення на екрані. Також на цьому етапі виконується відсікання невидимих поверхонь. Відсікання відкидає всі фрагменти, які знаходяться поза полем зору, підвищуючи продуктивність.

6. **Піксельний шейдер.** Основною метою піксельного шейдера є обчислення фінального кольору пікселя. Зазвичай піксельний шейдер містить дані про 3D-сцену, які він може використовувати для розрахунку остаточного кольору пікселя (наприклад освітлення, тіні, колір світла, тощо).

7. **Альфа-тест та змішування.** Після визначення всіх відповідних значень кольорів кінцевий об'єкт пройде ще один етап, який називається альфа-тестом і етапом змішування. На цьому етапі перевіряється відповідне значення глибини фрагмента, яке використовується, щоб перевірити, чи отриманий фрагмент знаходиться попереду чи позаду інших об'єктів. Етап також перевіряє альфа-значення (прозорість) і відповідно змішує кольори напівпрозорих об'єктів, які пересікаються.

OpenGL надає можливість контролювати графічний конвеєр з метою досягнення потрібного результату. Можливість створювати власні вершинні, геометричні та піксельні шейдери дозволяє відобразити будь-який об'єкт, що піддається математичному опису.

### Висновок

При роботі з комп'ютерною графікою виникають задачі, які вимагають великої кількості обчислень, що робить графічний процесор незамінним для виконання подібних завдань. OpenGL надає можливість контролювати графічний конвеєр та створювати власні вершинні, геометричні та піксельні шейдери для досягнення потрібного результату. Таким чином, OpenGL відкриває можливості для створення візуальних образів у широкому спектрі застосувань від ігор і анімацій до віртуальних середовищ та реалістичних симуляцій.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Комп'ютерна графіка : навчальний посібник / Романюк О.Н. Романюк О.В., Чехмєструк Р. Ю. – Вінниця: ВНТУ, 2022. – 141 с
2. OpenGL Programming Guide / John Kessenich, Graham Sellers, Dave Shreiner, 2016.
3. Романюк О. Н., Яковенко О. О., Романюк О.В., Котлик С. В Аналіз крос-платформового програмного інтерфейсу OPENGL і його нововведень. Інформаційні технології і автоматизація – 2021/ Матеріали XIV міжнародної науково- практичної Конференції . Одеса, 21-22 жовтня 2021р. -Одеса, Видавництво ОНАХТ, 2021 р. –С. 255-260.
4. LearnOpenGL: онлайн-посібник / Joey de Vries. URL: <https://learnopengl.com/>

**Іванчук Юрій Віталійович** – студент групи ІПІ-23м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: [ivanchukura.kunavi@gmail.com](mailto:ivanchukura.kunavi@gmail.com)

**Романюк Оксана Володимирівна** – к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com)

**Yuri Ivanchuk** – student of group ІPI-23m, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: [ivanchukura.kunavi@gmail.com](mailto:ivanchukura.kunavi@gmail.com)

**Oksana Romaniuk** – Candidate of Technical Sciences, Associate Professor of the Software Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com)