

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ ХАРЧОВИХ ТЕХНОЛОГІЙ
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ І ТЕХНОЛОГІЙ
«ІНДУСТРІЯ 4.0» ІМ. П.Н. ПЛАТОНОВА

**«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ І
АВТОМАТИЗАЦІЯ – 2021»**

***МАТЕРІАЛИ
XIV МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ***



21 - 22 ЖОВТНЯ 2021 р.

м.ОДЕСА

**MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
ODESSA NATIONAL ACADEMY OF FOOD TECHNOLOGIES
INSTITUTE OF COMPUTER SYSTEMS AND TECHNOLOGIES
"INDUSTRY 4.0" NAMED AFTER P.N. ПЛАТОНОВА**

**«INFORMATION TECHNOLOGIES AND
AUTOMATION– 2021»**

***PROCEEDINGS
OF THE XIV INTERNATIONAL SCIENTIFIC AND PRACTICAL
CONFERENCE***



OCTOBER 21 - 22, 2021

ODESSA

Організаційний комітет конференції
Organizational committee of the conference

Голова
Supervisor

Єгоров Б.В., проф. (Одеса)

Заступники голови
Deputy Chairmen

Поварова Н.М., доц. (Одеса, Україна)
Хобін В.А., проф. (Одеса, Україна)
Котлик С.В., доц. (Одеса, Україна)

Члени комітету
Committee members

Panagiotis Tzionas prof. (Thessaloniki, Greece)
Qiang Huang, prof. (Los Angeles C.A., USA)
Yangmin Li, prof (Macao, China)
Артеменко С.В., проф., (Одеса, Україна)
Романюк О.Н., проф. (Вінниця, Україна)
Грабко В.В., проф. (Вінниця, Україна)
Єгоров В.Б., д.т.н. (Одеса, Україна)
Жученко А.І., проф. (Київ, Україна)
Купріянов А.Б., доц. (Мінськ, Білорусія)
Ладанюк А.П., проф. (Київ, Україна)
Лисенко В.Ф., проф. (Київ, Україна)
Любчик Л.М., проф. (Харків, Україна)
Палов І., проф. (Русе, Болгарія)
Плотніков В.М., проф. (Одеса, Україна)
Стовкова В.Д., доц. (Тракия, Болгарія)
Суслов В., доц. (Кошалін, Польща)
Трішин Ф.А., доц. (Одеса, Україна)

УДК 004.01/08

Інформаційні технології і автоматизація – 2021 / Матеріали XIV міжнародної науково-практичної конференції. Одеса, 21-22 жовтня 2021 р. - Одеса, Видавництво ОНАХТ, 2021 р. – 350 с.

Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.

Збірник буде корисним як для фахівців і працівників фірм, зайнятих в області ІТ, так і для викладачів, магістрів і студентів вищих навчальних закладів, які навчаються за напрямками і спеціальностями програмного забезпечення обчислювальної техніки і автоматизованих систем, прикладної математики та обробки інформації, буде корисним професіоналам з комп'ютерного моделювання та розробки комп'ютерних ігор.

Результати досліджень у збірнику представляють собою своєрідний зріз сучасного стану справ в перерахованих галузях знань, який може допомогти як фахівцям, так і студентам університетів скласти загальну картину розвитку інформаційних технологій та пов'язаних з ними питань.

Наукові праці згруповані за напрямками роботи конференції та наведені в алфавітному порядку прізвищ авторів.

Матеріали (тези доповідей) друкуються в авторській редакції. Відповідальність за якість та зміст публікацій несе автор.

Рекомендовано для публікації Вченою Радою навчально-наукового інституту комп'ютерних систем і технологій «Індустрія 4.0» ім. П.М. Платонова від 23.09.2021 р., протокол № 2.

Матеріали подано українською, російською та англійською мовами.
Редактор збірника Котлик С.В.

На основі наведених у таблиці 1 отримано такі рівняння на основі логарифмічної функції

$$D_x = 5,4376 \cdot \ln(x) + 32,803,$$

$$D_{\bar{a}} = 5,4376 \cdot \ln(x) + 33,803.$$

Запропоновану методику можна використати і для оцінки вікових змін людини (рис. 5)



Рисунок 5– Морфінг зображення для оцінки вікових змін

Наведені пропозиції по використанню морфінгу зображень обличчя людини можуть бути використанні в медичній діагностиці та планування і проведення пластичних операцій.

Список використаної літератури

- [7] О.Н. Романюк, *Комп'ютерна графіка: навч. Посіб.* Вінниця : ВДТУ, 2001, 130 с.
- [8] О. Н. Романюк, та А. В. Чорний, *Високопродуктивні методи та засоби зафарбовування тривимірних графічних об'єктів. Монографія.* УНІВЕСУМ-Вінниця, Вінниця, 2006, 190 с.
- [9] А. Прохоров, Программы по морфингу на любой вкус. *КомпьютерПресс*, № 5, 2005

УДК 004.921

АНАЛІЗ КРОС-ПЛАТФОРМОВОГО ПРОГРАМНОГО ІНТЕРФЕЙСУ OpenGL I ЙОГО НОВОВВЕДЕНЬ

Романюк О. Н., Яковенко О. О., Романюк О.В., Котлик С. В. (rom8591@gmail.com, olesjayakovenko@gmail.com, sergknet@gmail.com)

*Вінницький національний технічний університет,
Одеська національна академія харчових технологій (Україна)*

Проведено аналіз крос-платформового програмного інтерфейсу OpenGL для розробки застосунків 2D і 3D комп'ютерної графіки. Наведено особливості використання, основні функції, графічні примітиви. Охарактеризовано основні нововведення.

Сьогодні тривимірну графіку використовують в різних галузях діяльності людини. Програмні засоби комп'ютерної графіки забезпечують формування високореалістичних зображень, створюючи віртуальну копію об'єктів. Раніше для кожної програми реалізовувалися власні методи, які використовувалися для візуалізації графічної інформації, оскільки не існувало єдиного стандарту в галузі графіки. Додатки писалися з використанням накопиченого досвіду. Сьогодні тривимірна графіка стала реальністю для персональних комп'ютерів, завдяки розробці потужних процесорів і графічних прискорювачів. Але через відсутність будь-яких стандартів, писати програми, незалежні від обладнання і операційної системи, було для виробників програмного забезпечення стало проблематичним. OpenGL є

OpenGL – це прикладний програмний інтерфейс з великим набором функцій, які використовуються для управління 2D і 3D графікою та зображеннями. Налічується понад 350 базових функцій основної бібліотеки `opengl32.dll`, які дозволяють формувати складні тривимірні сцени за допомогою примітивів (точки, відрізки, полігони), дають можливість обмежувати ділянки видимості, перетворювати координати, управляти освітленням, кольором, туманом, текстурою. В бібліотеці утиліт `glu32.dll`, які є розширенням базового набору функцій є функції для побудови зображень кубів, сфер, конічних циліндрів, дисків. Є можливість керування текстурою і перетворювати координати, триангулювати багатокутники, формувати криві та поверхні на нерегулярній сітці контрольних точок, застосовуючи форми Без'є та раціональних B-сплайнів [1].

Всі графічні двомірні та тривимірні об'єкти в OpenGL є примітивами (набір точок, ліній і багатокутників), які задаються вершинами. Є десять різних примітивів, за допомогою яких будуються всі об'єкти (рис. 1).

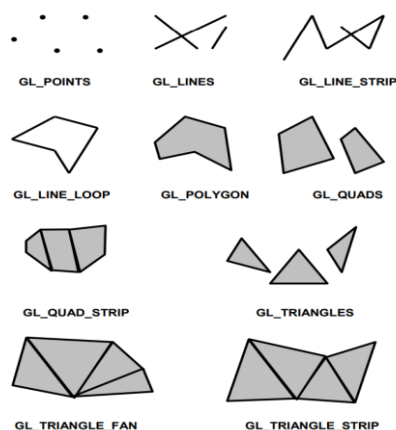


Рисунок 1 – Примітиви OpenGL

GL_POINTS – точка задається кожною вершиною,

GL_LINES – лінія задається кожною окремою парою вершин,

GL_LINE_STRIP – лінія задається кожною парою вершин (початком наступної лінії є кінець попередньої),

GL_LINE_LOOP – лінія задається кожною парою вершин (остання вершина з'єднується з першою, формуючи замкнуту фігуру),

GL_TRIANGLES – трикутник задається кожною окремою трійкою вершин,

GL_TRIANGLE_STRIP – трикутник задається кожною наступною вершиною разом з двома попередніми (формується стрічка з трикутників),

GL_TRIANGLE_FAN – перша вершина та наступні пари задають кожен трикутник (трикутники формуються навколо першої вершини та утворюють фігуру, подібну до діафрагми),

GL_QUADS – чотирикутник утворюється кожними чотирма вершинами,

GL_QUAD_STRIP – чотирикутник утворюється кожною наступною парою вершин разом з парою попередніх,

GL_POLYGON – задається багатокутник з кількістю кутів, що дорівнює кількості заданих вершин.

Функції поділяються на такі категорії:

– функції опису примітивів визначають об'єкти нижнього рівня ієрархії (примітиви), які здатна відображати графічна система. У OpenGL примітивами є точки, лінії, багатокутники і т.д;

– функції опису джерел світла служать для опису положення і параметрів джерел світла, розташованих у тривимірній сцені;

– функції завдання атрибутів. За допомогою завдання атрибутів програміст визначає, як будуть виглядати на екрані об'єкти, які відображаються. Тобто, якщо за допомогою примітивів визначається, що саме з'явиться на екрані, то атрибути визначають правила виводу на екран. Як атрибути OpenGL використовує колір, характеристики матеріалу, текстури, параметри освітлення;

– функції візуалізації дозволяють задати положення спостерігача у віртуальному просторі, параметри об'єктива камери. Знаючи ці параметри, система зможе не тільки правильно побудувати зображення, але і відсікти об'єкти, які не потрапили в поле зору;

– набір функцій геометричних перетворень дозволяє програмісту виконувати різні перетворення об'єктів – поворот, зсув, масштабування [1].

Архітектура OpenGL реалізує схему конвеєра, яка складається з декількох послідовних етапів обробки графічних даних (рис. 2).

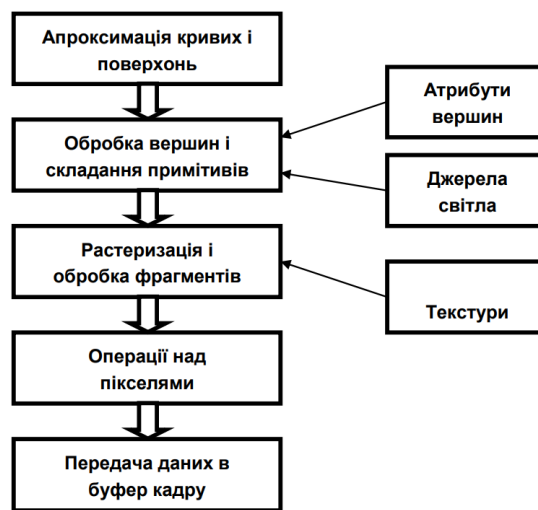


Рисунок 2 – Схема функціонування конвеєра OpenGL

OpenGL використовується індустрією комп'ютерних ігор, віртуальної реальності, при візуалізації наукових досліджень, в системах автоматизованого проектування, у графічних інтерфейсах тощо. OpenGL є специфікацією, яка розроблена і підтримується Khronos Group. Специфікація є документом, що описує набір функцій і їх точну поведінку, визначає, яким повинен бути результат кожної функції і як вона повинна виконуватися. Реалізація цієї специфікації вже залежить від конкретних розробників, тому версії OpenGL можуть мати різні реалізації. Бібліотеки OpenGL, зазвичай, розробляють виробники відеокарт.

Бібліотека OpenGL проста для використання, має широким спектр можливостей. Всі зміни, які вносяться до OpenGL, анонсують заздалегідь та реалізують так, щоб вже існуюче програмне забезпечення підтримувало нові графічні карти. Функції, які використовують OpenGL, не залежать від операційної системи чи устаткування та забезпечують однаковий візуальний результат. Додатки, які використовують відкриту графічну бібліотеку, можуть бути запущеними під різними операційними системами та на різних архітектурах. При перекомпіляції програми, OpenGL забезпечує портативність на рівні вихідних кодів [2]. Отже, перевагами OpenGL є:

- стабільність;
- надійність;
- портативність.

Бібліотека у розпорядженні розробників надає такі основні можливості [3].

- геометричні та растрові примітиви. На основі геометричних і растрових примітивів будуються всі об'єкти. З геометричних примітивів бібліотека надає: точки, лінії, полігони. З растрових: бітовий масив (bitmap) і образ (image);
- використання B-сплайнів. B-сплайни використовуються для того, щоб намалювати криві по опорних точках;
- видові і модельні перетворення. За допомогою цих перетворень можна розташовувати об'єкти в просторі, обертати їх, змінювати форму, а також змінювати положення камери з якої ведеться спостереження;
- робота з кольором. OpenGL надає програмісту можливість роботи з кольором в режимі RGBA (червоний-зелений-синій-альфа) або використовуючи індексний режим, де колір вибирається з палітри;
- видалення невидимих ліній і поверхонь. Z-буферизація;
- подвійна буферизація. OpenGL надає як одинарну так і подвійну буферизацію. Подвійна буферизація використовується для того, щоб усунути мерехтіння при мультиплікації, тобто зображення кожного кадру спочатку малюється в другому (невидимому) буфері, а потім, коли кадр повністю сформований, весь буфер відображається на екрані;
- накладання текстури. Дозволяє надавати об'єктам реалістичність. На об'єкт, наприклад, куля, накладається текстура, в результаті чого наш об'єкт тепер виглядає не просто як куля, а як різнобарвний м'ячик;
- згладжування. Згладжування дозволяє приховати ступінчастість, властиву растровим екранам. Згладжування змінює інтенсивність і колір пікселів близько лінії, при цьому лінія виглядає згладженою;
- освітлення. Дозволяє задавати джерела світла, їх розташування, інтенсивність, і т.д;
- атмосферні ефекти. Наприклад туман, дим. Все це дає підстави надати об'єктам або сцені реалістичність, а також "відчути" глибину сцени;
- прозорість об'єктів;
- використання списків зображень [3].

11 березня 2010 року Khronos Group представила фінальний варіант специфікації OpenGL 4.0 і мови обробки напівтонів GLSL 4.0. Режим сумісності введений в OpenGL 3.2, тому OpenGL 4.0 повністю сумісна з попередніми розширеннями. Специфікація OpenGL 4.0 була розроблена робочою групою OpenGL ARB (Architecture Review Board) в Кроносі (Khronos), і включає в себе оновлену мову GLSL 4.00 OpenGL Shading. Це дозволяє розробникам отримати доступ до останніх поколінь прискорювачів GPU зі значно поліпшеною якістю графіки, продуктивністю і збільшеною швидкістю гнучкого програмування (рис. 3) [4].

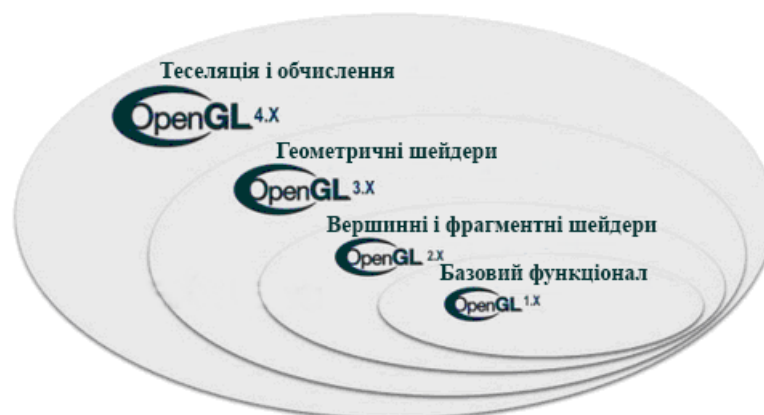


Рисунок 3 – Еволюція OpenGL

Нововведення:

- два нові етапи обробки шейдерів, що дозволяє перенести обробку тесселяції з центрального процесора на GPU;
- підтримка вибіркового фрагментних шейдерів (per-sample fragment shaders) і запрограмування вхідних позицій для збільшення якості рендеринга та гнучкості в управлінні згладжуванням фрагментних шейдерів;
- відображення даних, які згенеровані OpenGL або зовнішніми API, як OpenCL, без участі центрального процесора;
- шейдерні підпрограми допоможуть значно збільшити гнучкість написання програм;
- поділ стану текстур і текстурних даних через реалізацію нового типу даних під назвою "об'єкти семплів";
- 64-бітова подвійна точність з плаваючою комою операцій з шейдерами і введення-виведення для збільшення точності та якості рендеринга;
- збільшення продуктивності, а також шейдерів геометрії зразка, масивів зразка та нових запитів до таймеру [5].

Важливим нововведенням OpenGL 4.6 є можливість обробляти проміжне представлення шейдерів SPIR-V, який розроблений для API Vulkan. SPIR-V є універсальним для всіх платформ і може бути застосований не тільки для графіки, а і для паралельних обчислень. SPIR-V реалізує виділення окремої фази компіляції шейдерів в проміжне подання. Це дозволяє створювати фронтенди для різних високорівневих мов. Окремо генерується єдиний проміжний код на основі різних високорівневих реалізацій, що може бути використано драйверами OpenGL, Vulkan і OpenCL, не застосовуючи вбудованого компілятора шейдерів. Якщо вилучити драйвер від компілятора шейдерів, то можна помітити істотне спрощення драйвера, що прискорює завантаження коду для GPU і робить драйвер незалежним від високорівневих мов розробки програм для GPU. Для перетворення шейдерів на мові GLSL в формат SPIR-V розроблено компілятор glslang, в який вже додана підтримка GLSL 4.60 [6].

До основного складу специфікації OpenGL 4.6 включено 11 розширень:

- GL_ARB_gl_spirv і GL_ARB_spirv_extensions для стандартизації підтримки SPIR-V в OpenGL;
 - GL_ARB_indirect_parameters і GL_ARB_shader_draw_parameters для зниження навантаження на CPU при виконанні операцій в пакетному режимі, які пов'язані з рендерингом великого числа геометричних примітивів;
 - GL_ARB_pipeline_statistics_query і GL_ARB_transform_feedback_overflow_query для стандартизації в OpenGL раніше специфічних для Direct3D засобів для отримання статистики про хід виконання шейдерів і відслідження переповнення буферів;
 - GL_ARB_texture_filter_anisotropic (раніше розширення GL_EXT_texture_filter_anisotropic) з реалізацією методу покращення візуальної якості текстур, на який раніше поширювалися патентні обмеження;
 - GL_ARB_polygon_offset_clamp (раніше розширення GL_EXT_polygon_offset_clamp) для усунення візуального артефакту "витік світла" (light leak), який виникає при відображенні тіней;
 - GL_ARB_shader_atomic_counter_ops і GL_ARB_shader_group_vote з реалізацією додаткових функцій шейдерів, що розширюють функціональність і продуктивність рішень для робочого столу (реалізація атомарних лічильників і функції для прискорення композітінга на процесорах SIMD);
 - GL_KHR_no_error, дозволяє знизити навантаження на драйвер при виконанні операцій. За допомогою даного розширення додаток може відключити код перевірки помилок в драйвері, що позитивно впливає на продуктивність.
- Додано три нововведення, які в подальшому будуть оформлені як розширення OpenGL:
- GL_KHR_parallel_shader_compile – дозволяє додаткам запуснути відразу декілька потоків компіляції шейдерів;

– `WGL_ARB_create_context_no_error` і `GXL_ARB_create_context_no_error` для відключення контексту обробки помилок в WGL або GLX.

Для покращення сумісності з API Vulkan і Direct3D в специфікації OpenGL і OpenGL ES додана частина необов'язкових розширень для низькорівневих маніпуляцій з об'єктами в пам'яті і управління синхронізацією виконання операцій з GPU: `GL_EXT_memory_object` * і `GL_EXT_semaphore` *. Зазначені розширення дають можливість імпортувати в OpenGL-додатки об'єкти Vulkan для їх прив'язки до текстур або буферам в пам'яті. Для сумісності з Direct3D додано розширення `GL_EXT_win32_keyed_mutex`.

Специфікація OpenGL широко застосовується індустрією комп'ютерних ігор і віртуальної реальності, у графічних інтерфейсах (Comviz, Clutter), при візуалізації наукових даних, в системах автоматизованого проектування тощо.

Список використаної літератури

1. Геометричне моделювання і комп'ютерна графіка: використання бібліотеки OpenGL / А. А. Лященко, В. В. Демченко, Є. В. Бородавка, В. В. Смирнов. – Київ: Київський національний університет будівництва і архітектури, 2008. – 92 с.
2. OpenGL: Open Graphics Library [Електронний ресурс] // gamedev. – 2018. – Режим доступу до ресурсу: <http://www.gamedev.ru/terms/OpenGL> (Дата звернення 05.10.2021)
3. Що таке OpenGL? [Електронний ресурс] // 3dnews. – 2000. – Режим доступу до ресурсу: <https://3dnews.ru/169184> (Дата звернення 05.10.2021)
4. Трішки про OpenGL [Електронний ресурс] // Анві. – 2017. – Режим доступу до ресурсу: <http://esate.ru/uroki/OpenGL/uroki-OpenGL-c-sharp/chto-takoe-opengl> (Дата звернення 05.10.2021)
5. OpenGL [Електронний ресурс]. Режим доступу до ресурсу: <https://znaimo.com.ua/OpenGL> (Дата звернення 05.10.2021)
6. Випуск специфікації OpenGL 4.6 [Електронний ресурс] // Опеннет. – 2017. – Режим доступу до ресурсу: <https://www.opennet.ru/opennews/art.shtml?num=46952> (Дата звернення 05.10.2021)

УДК 519.6+659

РОЗРОБКА МОДУЛЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ УПРАВЛІННЯ РЕКЛАМНОЮ КАМПАНІЄЮ ТОРГІВЕЛЬНОГО ПІДПРИЄМСТВА

Рябікіна Д.О., Шевченко Н.Ю. (rabikinadara3@gmail.com, natalygustav@gmail.com)

Донбаська державна машинобудівна академія (Україна)

В тезах зазначено, що проблема управління рекламною кампанією з урахуванням впливу конкуренції на ринку є актуальною і потребує вирішення, в тому числі через розробку відповідного математичного и технічного інструментарію. Авторами наведений математичний алгоритм оптимального розподілу бюджету рекламної кампанії між видами рекламних заходів торговельного підприємства з позиції максимізації прогнозованої виручки від реалізації продукції. Наведена діаграма варіантів використання та приклади роботи програмного модуля, що реалізує запропонований математичний алгоритм.

На сучасному етапі в процесі здійснення торговельними підприємствами рекламної діяльності спостерігається необґрунтованість вибору рекламних носіїв, відсутність досліджень при створенні та впровадженні рекламних кампаній, неефективність витрат на рекламу. Тому проблема управління рекламною кампанією з урахуванням впливу конкуренції на ринку є актуальною і потребує вирішення, в тому числі через розробку відповідного математичного и технічного інструментарію.