

різних архітектур для реалізації моделювання та прогнозування показників ефективності управління інформаційно-інтелектуальною системою з прогнозуванням якості харчової продукції. Перевага нейромережних методів в тому, що вони дають змогу відтворювати складні нелінійні залежності й виконувати прогноз управління процесами на довгий період з високою ймовірністю результатів.

Список літературних джерел

1. Субботін С.О. Нейронні мережі: теорія та практика: навч. посіб. / С. О. Субботін. – Житомир: Вид. О. О. Євенок, 2020. – 184 с.
2. Тимошук П.В. Штучні нейронні мережі: [навч. посіб.] / П.В.Тимошук. - Львів: Видавництво Львівської Політехніки, 2011. – 444 с.
3. Chubaievskyi, V., Lakhno, V., Kryvoruchko, O., Kasatkin, D., Desiatko, A., Blozva, A., & Gusev, B. (2021). Методика мінімізації витрат на побудову багатоконтурної системи захисту на основі генетичного алгоритму. Електронне фахове наукове видання "Кібербезпека: освіта, наука, техніка" 1(13), 16-28.
4. Криворучко О.В., Костюк Ю.В., Самойленко Ю.О., Савчук О.В. Сучасні комп'ютерні технології для статистичних методів управління якістю // Sciences of Europe (Praha, Czech Republic). VOL 1, N68.- (2021) —с. 51-55.

УДК 004.921

*Крохмалюк В. В., здобувач освіти 4 курсу спеціальності 122 «Комп'ютерні науки»
Римар П. В., старший викладач кафедри інформаційних технологій*

ВИКОРИСТАННЯ КРОС-ПЛАТФОРМНОГО ПРОГРАМНОГО ІНТЕРФЕЙСУ OPENGL В КОМП'ЮТЕРНІЙ ГРАФІЦІ

Донецький національний університет імені Василя Стуса, м. Вінниця

OpenGL перекладається як Відкрита Графічна Бібліотека (Open Graphics Library), це означає, що OpenGL – це відкритий та мобільний стандарт. Програми, написані за допомогою OpenGL, можна переносити практично на будь-які платформи, отримуючи при цьому однаковий результат, будь то графічна станція або суперкомп'ютер. OpenGL звільняє програміста від написання програм конкретного устаткування. Якщо пристрій підтримує якусь функцію, то ця функція виконується апаратно, якщо ні – бібліотека виконує її програмно.

З точки зору програміста OpenGL – це програмний інтерфейс для графічних пристроїв, таких як графічні прискорювачі. Він включає близько 150 різних команд, за допомогою яких програміст може визначати різні об'єкти і виробляти рендеринг. Говорячи простішою мовою, ви визначаєте об'єкти, задаєте їх місце розташування в тривимірному просторі, визначаєте інші

параметри (поворот, масштаб, та ін), задає властивості об'єктів (колір, текстура, матеріал, та ін), положення спостерігача, а бібліотека OpenGL подбає про те, щоб відобразити все це на екрані. Тому можна сказати, що бібліотека OpenGL є лише відтворюючою (Rendering) і займається лише відображенням 3D об'єктів, вона не працює із пристроями введення (клавіатури, миші). Також вона не підтримує менеджера вікон. OpenGL має добре продуману внутрішню структуру та досить простий процедурний інтерфейс. Незважаючи на це, за допомогою OpenGL можна створювати складні та потужні програмні комплекси, витрачаючи при цьому мінімальний час у порівнянні з іншими графічними бібліотеками. Як приклад цьому можна навести результат виводу досить невеликої програми (рис. 1).

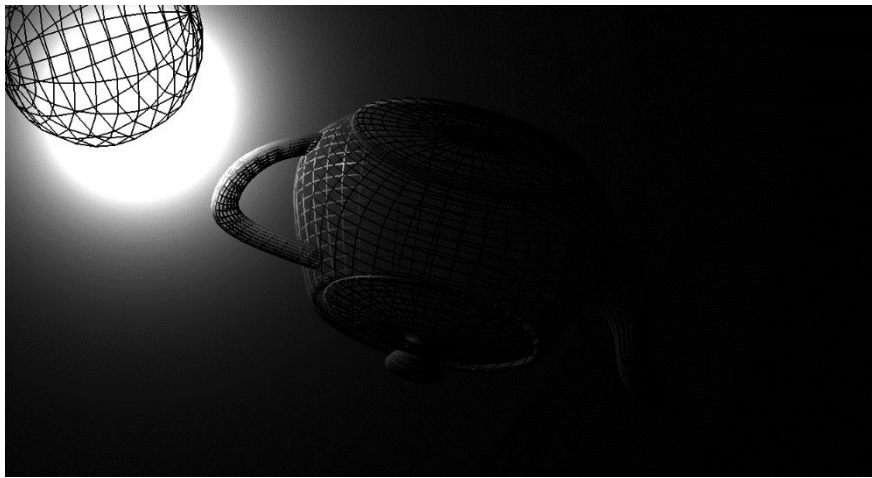


Рисунок 1 – Приклад роботи програми, написаної за допомогою графічної бібліотеки OpenGL

Відмінною особливістю OpenGL є підтримка розширень. Щоразу, коли графічна компанія випускає нову методику або нову велику оптимізацію для рендерингу, це часто зустрічається у розширенні, реалізованому у драйверах. Якщо обладнання, на якому працює програма, підтримує таке розширення, розробник може використовувати функціонал, що надається цим розширенням, для більш просунутої або ефективної графіки. Таким чином, графічний розробник вже може використовувати нові методи рендерингу, просто перевіряючи, чи підтримується дане розширення відеокартою, при цьому не чекаючи поки OpenGL додасть цей функціонал у свою нову версію. Часто, коли розширення є популярним або дуже корисним, воно зрештою стає частиною нової версії OpenGL.

OpenGL сам по собі є великою системою станів, яка містить цілий набір змінних, що визначають те, як OpenGL повинен зараз виконувати операції. Стан OpenGL зазвичай називають контекстом OpenGL. При використанні OpenGL часто змінюватиметься його стан, встановлюючи деякі параметри, маніпулюючи значеннями з буфера, а потім виконуючи рендеринг з використанням поточного контексту.

Наприклад, якщо зараз OpenGL малює трикутники, а потрібні лінії, то можна змінити стан OpenGL, переглядаючи деяку змінну контексту, яка

встановлює спосіб малювання OpenGL. Як тільки контекст буде змінено, повідомивши OpenGL, що він повинен малювати лінії замість трикутників, всі наступні команди малювання будуть вже автоматично малювати лінії (а не трикутники).

Під час роботи з OpenGL стикаємося з певними функціями зміни стану, які змінюють контекст, та певними функціями використання стану, які виконують зазначені операції, залежно від поточного стану OpenGL.

OpenGL було розроблено з урахуванням кількох абстракцій. Однією із таких абстракцій є об'єкти. Об'єкт у OpenGL – це набір параметрів, що є підмножиною стану OpenGL. Наприклад, може бути об'єкт, який представляє параметри вікна малювання; тоді є можливість змінити його розмір, кількість кольорів, що підтримуються, і так далі. Об'єкт OpenGL можна собі візуалізувати як звичайну структуру.

Перевагою використання об'єктів є те, що наявна можливість визначити більше одного об'єкта в додатку, встановити їх параметри, і щоразу, коли запускається операція, яка використовує стан OpenGL, пов'язується об'єкт з кращими налаштуваннями. Наприклад, є об'єкти-контейнери для зберігання даних 3D-моделі (Будинок або Персонаж), і щоразу, коли потрібно намалювати 3D-модель, відбувається пов'язування з необхідним контекстом об'єкт, що містить дані моделі, яку потрібно намалювати (перед цим створивши та встановивши параметри для цих об'єктів). Наявність декількох об'єктів дозволяє вказувати безліч моделей, і щоразу, коли потрібно намалювати конкретну модель, перед малюванням потрібно просто пов'язати відповідний об'єкт (не встановлюючи йому всі параметри з нуля) з необхідним контекстом.

Також не менш важливим є те, що це «відкритий стандарт». Будь-яка компанія, що має апаратну платформу, може купити ліцензію у SGI і зробити власну реалізацію OpenGL. Зміни в OpenGL пропонуються, обговорюються та затверджуються представниками різних компаній.

У висновку можна сказати, що OpenGL має добре продуману внутрішню структуру та досить простий процедурний інтерфейс. Незважаючи на це, за допомогою OpenGL можна створювати складні та потужні програмні комплекси, витрачаючи при цьому мінімальний час у порівнянні з іншими графічними бібліотеками.

Список використаної літератури

1. Gabor Szauer, Hands-On C++ Game Animation Programming: Learn modern animation techniques from theory to implementation with C++ and OpenGL, 2020 – 368ст.
2. de Vries, Joey, Learn OpenGL: Learn modern OpenGL graphics programming in a step-by-step fashion, 2020 – 522ст.
3. Sumanta Guha, Computer Graphics Through OpenGL: 3 Theory to Experiments, 2019 – 761ст.