

АНІМАЦІЯ РУХУ МОДЕЛІ СУПУТНИКА ІЗ QT QUICK 3D API

Космічна галузь має максимально ефективно використовувати переваги сучасних технологій, серед яких є і 3D-анімація, щоб задовольняти постійно зростаючим вимогам до якості програмного та апаратного забезпечення космічних апаратів (КА), а також спрощувати процес їхньої експлуатації.

У рамках роботи над створенням моделюючого програмного забезпечення (ПЗ) симулятора для відпрацювання системи керування (СК) КА, окрім відображення числових результатів моделювання та їх інтерпретації у двовимірні графіки залежності, виникла необхідність у демонстрації руху 3D-моделі супутника відносно відповідних просторових осей (x , y , z).

Зважаючи на те, що в роботі вже використовувався фреймворк Qt, було розглянуто варіанти сумісних з Qt платформи-незалежних API для впровадження 3D-графіки – OpenGL та Qt Quick 3D.

OpenGL є низькорівневим процедурним API, який втілює імперативний підхід до створення графіки. З одного боку, це вимагає глибокого знання законів 3D-графіки і математичних моделей, з іншого боку – дає свободу впровадження різних інновацій. Проте розширення функціоналу стандарту OpenGL сторонніми бібліотеками полегшує і прискорює процес створення тривимірної сцени [1]. Наприклад, Qt Quick 3D, поєднуючи в собі QML / C++ API та маючи за основу OpenGL для рендерингу (візуалізації об'єктів), є модулем Qt для розробки програм з 3D-графікою.

Qt Quick 3D має низку переваг у порівнянні з «чистим» OpenGL: по-перше, це нативність даного API для проекту на Qt, по-друге, його актуальність й активне вдосконалення (останні зміни були внесені у 2019 р., для OpenGL – у 2017 р.), і, нарешті, простота у розумінні та використанні його головних властивостей – «Entity-Component-System» (систему складають сутності, які утворені з компонентів (примітивів) із властивостями) та «Task-Based Engine» (рушій Qt Quick 3D опитує кожен кадр на набір завдань для виконання разом із залежностями між ними) [2].

Основа для Qt Quick 3D становлять скрипти, написані декларативною мовою QML, які можуть містити вставки коду на JavaScript. Мова QML добре оптимізована, має метаоб'єктну модель і зручна у взаємодії з кодом на C++ (метаоб'єктний або об'єктно-орієнтований сигнально-слотовий зв'язок).

Перевагою QML також є те, що за замовчуванням вона є інтерпретованою, тобто нема потреби компілювати її в бінарний код для запуску, але це можна зробити з метою підвищення швидкості виконання коду, використавши Qt Quick Compiler [3].

Qt Quick 3D надає достатній набір визначених моделей даних і видів їх представлення. Так, наприклад, клас *QQuaternion* дає змогу створювати кватерніони для подання положення та куту повороту об'єкту в тривимірному просторі [4].

Щодо шляхів використання даного модуля, то можливо з початку розробки в IDE Qt Creator вибрати опцію «Qt Quick Application». Таким чином, весь проект буде заснований здебільше на QML, ніж на C++. Однак, такий спосіб підходить більше для створення мобільних та веб-додатків.

Для моделюючого ПЗ симулятора обраний інший спосіб впровадження технології Qt Quick, а саме – використання класу *QQuickWidget*, який базується на класі *QWidget*, тобто поводження з ним у коді таке ж зручне, як й з іншими віджетами. Однак, коли застосовується графіка із складною динамікою, такий підхід може передбачати зниження продуктивності ПЗ [5].

Для завантаження опису геометрії (mesh) 3D-моделей найчастіше використовують текстовий формат OBJ, тому ще одним плюсом на користь Qt Quick 3D є реалізація в ньому автоматизованого читання OBJ-файлів в той час, як OpenGL потребує власноручного написання коду для цього. Даний формат легко імпортується та експортується у всі популярні середовища створення 3D-графіки, серед яких було обрано «Blender» (вільно поширюване та невибагливе до системних ресурсів).

Отже, використання API високого рівня – Qt Quick 3D – дозволило за короткій термін доповнити ПЗ симулятора візуалізацією руху 3D-моделі супутника. У майбутньому підхід до опрацювання 3D-контенту може видозмінюватися і покращуватися у зв'язку з розвитком можливостей Qt.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Khronos OpenGL Registry [Електронний ресурс]. – Режим доступу: https://www.khronos.org/registry/OpenGL/index_gl.php
2. S. Harmer Overview of Qt3D 2.0 – Part 1. [Електронний ресурс] / S. Harmer – Access mode: <https://www.kdab.com/overview-qt3d-2-0-part-1>
3. Lee Zhi Eng Hands-On GUI Programming with C++ and Qt / Lee Zhi Eng – UK : Packt Publishing, 2018 – 235 p.
4. Офіційна сторінка Qt Documentation [Електронний ресурс] – Режим доступу: <https://doc.qt.io>
5. Шлее М. Qt 5.10. Профессиональное программирование на C++. / М. Шлее – СПб : БХВ-Петербург – 2018 – 328 с.