

Спосіб організації базових елементів SIM-архітектури програмних засобів моделювання фізичних процесів

В.С.Семенякін, к.т.н. Т.М.Заболотня

Національний технічний університет України «КПІ»

Україна, Київ

semenyakinVS@rambler.ru

На сьогоднішній день однією з основних тенденцій розвитку сфери комп'ютерних анімації та ігор є постійно зростаюча потреба у програмних комплексах моделювання фізичних процесів (далі – *фізичних движках*). Їх використання дозволяє значно спростити процес створення мультиплікації та віртуальних світів комп'ютерних ігор, адже автоматизована реалізація зображень знімає необхідність відтворення різних фізичних явищ вручну.

Незважаючи на бурхливий розвиток галузі розробки фізичних движків і велику кількість вже існуючих доробків, тут можна виділити декілька аспектів, що потребують переосмислення та модифікації. Так, наразі переважна більшість сучасних програмних комплексів моделювання фізичних процесів має статичну архітектуру, яка не підтримує (або робить складною) можливість підключення бібліотек обробки даних щодо фізичних законів або колекцій законів з цілих розділів фізики. Іншою проблемою є відсутність засобів створення структур даних та алгоритмів для визначення відношень між фізичними об'єктами в середині системи, що моделюється. Користувачі движку мають власноруч описувати закони фізики, відтворюючи складну поведінку об'єктів простору моделювання за допомогою використання множини елементарних функцій впливу на ці об'єкти.

Таким чином, актуальним є пошук нових підходів до побудови архітектури фізичних движків. SIM-підхід, про який ідеться у доповіді, базується на об'єктоорієнтованій парадигмі програмування (ООП), ідеї предметноорієнтованого підходу до програмування (*subject-oriented programming*) та використанні шаблону проектування *стратегія*. Головна ідея полягає у виділенні методів і властивостей об'єктів простору моделювання в окремі програмні сутності (в ООП – класи) [1]. Внаслідок цього центральним елементом SIM-архітектури є головний клас-оболонка – *shell-class* (у нашому випадку – фізичний об'єкт), що виступає у ролі контейнеру, де зберігаються списки класів-властивостей (*model-class*) та класів-методів (*influence-class*), які реалізують вплив на стан заданого об'єкту.

У доповіді запропоновано спосіб модифікації вищезгаданої архітектури з точки зору вдосконалення логіки відношень між об'єктами. В результаті її складові елементи реалізуються так:

1. *Shell-class* (фізичний об'єкт) тепер містить лише посилання на властивості (*model-class*).
2. *Model-class-u* організовані як класи-контейнери, що зберігають списки властивостей з певного розділу фізики та містяться у глобальному кореневому класі – *фізичній системі*.
3. *Influence-class-u* відтепер напряму взаємодіють з *model-class-ами*. Список впливів (*influence*) на простір моделювання можна зберігати у фізичній системі або у *model-class-ax*.

Описаний спосіб організації базових елементів SIM-архітектури передбачає об'єднання властивостей фізичних об'єктів у групи і тим самим сприяє спрощенню алгоритмів роботи з об'єктами, а також підвищує зручність використання самого движку, оскільки виділення реалізації впливів (*influence*) з класів об'єктів робить організацію архітектури більш логічною.

Таким чином, описана модифікація архітектури фізичних движків сприяє зменшенню часу створення базових елементів останньої та дозволяє розділити процес програмного моделювання різних розділів фізики на декілька незалежних потоків. Завдяки цьому з'являється можливість колективної open-source розробки модульного програмного комплексу моделювання фізичних процесів.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Семенякін В.С, Заболотня Т.М. Узагальнений алгоритм моделювання фізичних процесів на базі мови програмування С# // "Прикладна математика та комп'ютеринг ПМК-2010": Зб.тез доповідей.— К.:Просвіта.— 2010.—с.353-356.