УДК 004.421

К.т.н., старший викладач Заболотня Т.М., студент Семенякін В.С.

**Національний технічний університет України
«Київський політехнічний інститут»**

УЗАГАЛЬНЕНИЙ АЛГОРИТМ МОДЕЛЮВАННЯ ФІЗИЧНИХ ПРОЦЕССІВ НА БАЗІ МОВИ ПРОГРАМУВАННЯ C#

Abstract

Tetiana M. Zabolotnia, snr. teacher, PhD; Semenyakin Volodymir, student
Generalized algorithm for physical processes modelling based on programming language C#.

The APE system of physical processes' modelling is reviewed as a comparison. The software system of physical processes modelling with a possibility of flexible enhancement and modification is developed. The ways for further modification are proposed as well.

Вступ

Усі сучасні комп'ютерні ігри у тій чи іншій мірі використовують комп'ютерне моделювання природних явищ для реалістичного відтворення навколишнього світу. У науці моделювання фізичних процесів використовується також набагато частіше, ніж реальні наукові експерименти. Отже, обчислення параметрів фізичних процесів є актуальною задачею як у науці, так і у галузі електронних розваг.

Дана робота присвячена питанню розробки узагальненого алгоритму обчислення характеристик фізичних процесів і явищ, що дозволяє гнучко керувати набором діючих на об'єкт законів і легко розширювати перелік цих законів.

Постановка задачі

Метою даної роботи є:

- реалізація можливості модифікації фізичних законів користувачем; підтримка доповнення існуючої структури новими фізичним законами, а також новими даними на обчислення;
- розробка інтуїтивно зрозумілої структури класів для програми реалізації комп'ютерного моделювання, розрахованої на подальше використання як OpenSource;

- забезпечення гнучкого інтерфейсу користувача-програміста.

Термінологія

Комп'ютерне моделювання фізичних процесів – це методи аналізу реальних чи очікуваних фізичних процесів за допомогою ЕОМ, коли ці процеси моделюються згідно даній послідовності фізичних механізмів [1].

UML (англ. Universal modeling language) - це мова позначень або побудови діаграм, призначена для визначення, візуалізації і документування моделей, зорієнтованих на об’єкти систем програмного забезпечення [2].

APE (англ. Actionscript Physics Engine) – безкоштовний двовимірний OpenSource движок для Flash/Flex, написаний мовою ActionScript [3].

Опис алгоритму

На даний час існує багато способів побудови алгоритмів, що реалізують такі обчислення. Але більшість з них ставить акцент на математичному розрахунку колізій та ізолюванні користувача алгоритму від тонкого керування силами і законами. Наприклад, у OpenSource системі фізичного моделювання APE [3] користувач не має доступу безпосередньо до самих фізичних законів. Він лише задає характеристики об'єктам і загальну стратегію розрахунку фізичних характеристик. Такий спосіб моделювання зручний лише до тих пір, доки відсутня необхідність обчислювати закони з нового напряму фізики. Структура APE не дозволяє зробити це з достатньо гнучко.

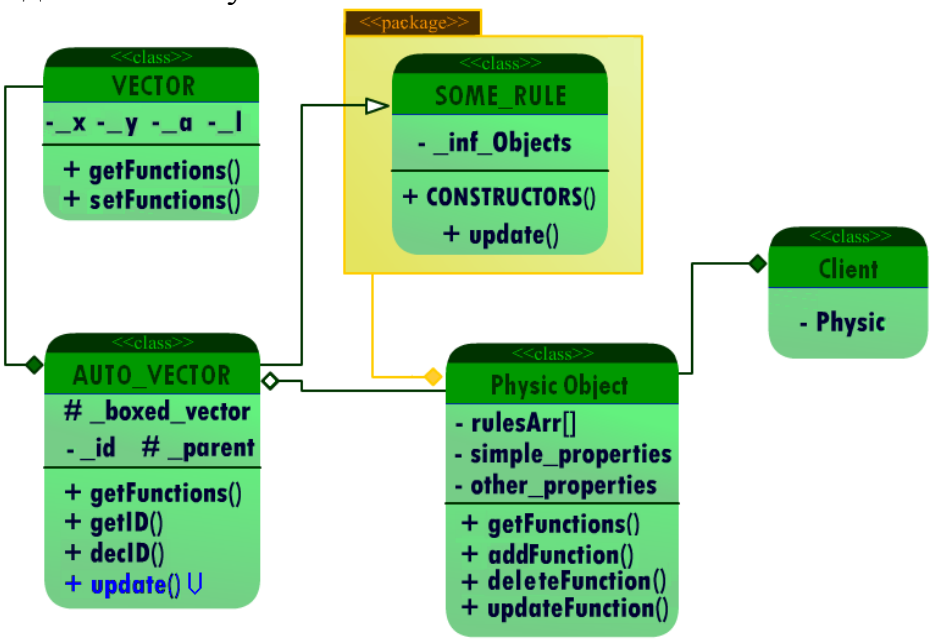

Рис.1. UML-схема системи загального моделювання фізичних процесів

Розглянемо структуру класів загальної фізичної моделі (див.рис.1).

Особливість нового узагальненого алгоритму полягає у абстрагуванні самого поняття «фізичний закон». Закон у цій роботі розглядається лише як деякий алгоритм, що змінює характеристики підлеглого йому тіла.

У вказаній структурі центральне місце займає абстрактний клас AUTO_VECTOR. Він служить обгорткою навколо _boxedVector - екземпляру класу CFVector. boxedVector - фізичний вектор сили, що контролюється даним законом. getFunctions() – набір функцій, що служать для обмеження доступу до полів _boxedVector, дозволяючи лише їх читання.

Віртуальна функція update() використовується у нащадках класу AUTO_VECTOR для обчислення вже конкретного фізичного закону.

Інформація про функції, пов'язані з полем _id, а також про поля _parent та _id, подана дещо нижче.

Класи вигляду SOME_RULE є реалізаціями деяких фізичних законів. У схемі вони зображені у вигляді набору класів package. Ці класи реалізують віртуальну функція update(), кожен по-своєму. Update() виконує безпосередньо обчислення закону.

Також у класі SOME_RULE задаються вказівники на об'єкти існуючої фізичної моделі, що мають вплив на даний закон. У UML-схемі вони вказані як _inf_Objects.

Клас physicObject – це клас, що виконує обчислення масиву включених у нього законів. Це реалізується через функцію updateFunction() наступним чином:

1. Динамічний масив законів rulesArr[] складається з екземплярів AUTO_VECTOR, що гарантують наявність функції update(). Ця функція викликається у циклі для всіх включених у даний physicObject законів. Сили, що повертаються кожним викликом, складаються й повертаються у поле _forceSum (у схемі це поле включене до групи simple_properties).
2. Після отримання суми сил через другий закон Ньютона обчислюється вектор прискорення тіла, що складається з вектором швидкості. Вектор швидкості додається до вектора положення тіла у просторі і положення повертається у якості результату.

Нові закони додаються до physicObject через метод addFunction(). У якості параметру ця функція приймає екземпляр нащадку AUTO_VECTOR.

Функція deleteFunction() приймає у якості параметру індекс закону (null-based), який необхідно виключити з масиву rulesArr[]. Як правило, ця функція використовується законом у випадку, якщо зникає хоча б один об'єкт впливу на нього.

Група властивостей simple_properties класу physicObject включає в себе прості динамічні характеристики (такі, як положення у просторі, швидкість, прискорення, суперпозиція сил і маса).

Група властивостей other_properties класу physicObject може зберігати специфічні характеристики, такі, наприклад, як кінетична чи потенційна енергії.

У сенсі аналізу класу physicObject слід розглянути також поля _parent і _id з класу AUTO_VECTOR. Вони були створені для зручної роботи з об'єктом physicObject:

_parent. Вказівник, через який екземпляри нащадків класу AUTO_VECTOR можуть отримувати доступ до physicObject, у який вони вкладені.

_id. Службовий параметр. Номер екземпляру даного закону у динамічному масиві rulesArr[]. Необхідний для коректного видалення з масиву.

Для більш зручного використання алгоритмів отриманої системи екземпляр класу physicObject треба включити до класу користувача, для якого необхідне фізичне моделювання. У конструкторі цього класу можна задати екземпляри сил з набору package і включити їх у physicObject даного класу через функцію addFunction(). Це буде коректно з точки зору логіки побудови коду.

Висновки

Таким чином, у даній роботі запропоновано новий узагальнений алгоритм комп’ютерного моделювання фізичних явищ та процесів. Відмінною рисою, що відрізняє його від вже існуючих подібних рішень, є те, що в ньому фізичний закон подається, як деякий алгоритм, що змінює характеристики підлеглого йому тіла. Це дозволило реалізувати можливість динамічної модифікації фізичних законів програмістом, а також введення нових законів до програми.

У подальшому запропоновану структуру класів можна модифікувати для зручної роботи з характеристиками газів, рідин, тощо.

Література

1. *Кривобокова Е.В.* «Элективный курс "Компьютерное моделирование физических процессов" для 10-11 классов» [Електронний ресурс http://pedsovet.su/load/70-1-0-2337], дата візиту 09.03.2010

2. Основи UML. [Електронний ресурс http://docs.kde.org/stable/uk/kdesdk/umbrello/uml-basics.html#about-uml], дата візиту 09.03.2010

3. APE (Actionscript Physics Engine) [Електронний ресурс http://www.cove.org/ape/], дата візиту 09.03.2010

4. *Judith Bishop* «C# 3.0 Design Patterns» // O'Reilly Media Inc, 2008.