Секція: Фізико-математичні науки

*Богонос В.О.*

*здобувач фахової передвищої освіти*

*Краматорського фахового коледжу промисловості, інформаційних технологій та бізнесу Донбаської державної машинобудівної академії*

*Новікова Н.В.*

*викладач-методист*

*Краматорського фахового коледжу промисловості, інформаційних технологій та бізнесу Донбаської державної машинобудівної академії*

**Роль математики у створенні комп`ютерних ігор**

Застосування комп’ютерних ігор є невід`ємною частиною сучасного життя. Здавалося б, до чого тут математика? Адже програмування – це мови, програмні коди, тощо. Давайте трохи розберемося.

По-перше, математика вчить абстрактному мисленню, розуміти та ставити завдання, виконувати різні дії і операції, аналізувати можливі рішення, вирішувати поставлені проблеми.

По-друге, тому що саме програмування і все пов'язане з комп'ютерами працює за рахунок цієї самої математики. Найпростіші програми й взагалі, обчислювальна робота комп'ютера, працює і ґрунтується на принципах математики, починаючи з найпростіших математичних операцій і виразів і закінчуючи складними обчисленнями.

Якщо гра є великим проектом, то її створення вимагає від розробників як мінімум знань з обчислювальної геометрії, аналітичної геометрії, чисельних методів, особливо з упором на метод кінцевих різниць і метод кінцевих обсягів, теорії ймовірностей, інтегрального і диференціального числення, теорії ігор і теорії прийняття рішень (для впровадження штучного інтелекту в проект).

Деякі речі, які зустрічаються при створенні ігор та повністю спираються на математику, це симуляція рідин, анімація, алгоритми, архітектура ігрових двигунів, написання ігрової логіки, аналітика і збір даних, розрахунок кадрів на секунду, ігрова фізика, графіка/шейдери, штучний інтелект, процедурна генерація, рендеринг полігонів і багато чого іншого ...

Найважливішими поняттями для розробки ігор з точки зору математики є вектори, дії з ними та скалярний і векторний добуток, скалярні маніпуляції, рух (особливо паралельне перенесення), матриці, тригонометрія. У великих сучасних проектах, як правило, використовується усе відразу.

Одним з напрямків в лінійної алгебри є вивчення векторів. В іграх векторивикористовуються для зберігання розташування, напрямків, швидкостей та прискорення. Якщо у грі застосовується позиціонування екранних кнопок, робота з камерою і її напрямком, швидкостями об'єктів, то обов`язково доведеться мати справу з ними. Чим краще розробник розуміє лінійну алгебру, тим більший контроль він отримує над поведінкою векторів і, отже, над самою грою.

Поняття «базисний вектор» дає можливість застосовувати перетворення зразу для цілого об’єкту. Це зручно, оскільки позбавляє від необхідності застосовувати тригонометричні перетворення до кожної з точок моделі об`єкту.

В іграх часто буває потрібно помножити вектор на число (скаляр). Наприклад, моделюючи простий опір повітря. А кут між векторами може стати в нагоді для визначення чи знаходиться об'єкт в зоні, яку ми бачимо. Такі процеси, як створення векторів, названих MAX\_SPEED (максимальна швидкість) і MIN\_SPEED (мінімальна швидкість), додають в ігровий цикл для перевірки та розрахунку швидкості об`єкту, деформації поверхонь при зіткненні і зміні траєкторії і швидкості в залежності від сили удару.

Ще одна область математики, яка займається дискретними структурами (наприклад: графами, автоматами, твердженнями в логіці) - це дискретна математика. Основна її відмінність від звичайної математики – її об'єкти не можуть змінюватися так само гладко, як і речові числа. Тому в якомусь сенсі всі завдання, які вирішуються в програмуванні, так чи інакше відносяться до дискретної математики, тому її знання стає в нагоді чи не будь-якому розробнику ігор. Наприклад, графи – це дуже зручні формалізовані уявлення нелінійних структур, які досить часто зустрічаються в прикладних задачах.

В основі всіх комп`ютерних науках лежить математична логіка. Але не потрібно заглядати глибоко в теорію, щоб знайти застосування цієї науки в написанні програм, та й взагалі в рішенні задач. Добре, якщо у програміста досить умінь, щоб написати рішення задачі. Але також важливо розуміти, яким чином можна довести, що код працює правильно. Більшість програм вирішує будь-яку математичну задачу, і потрібно вміти доводити, що завдання виконано правильно. Тоді на допомогу приходять методи логіки. Часто формальними перетвореннями можна скоротити обсяг коду і зробити його більш читабельним. А крім того, деякі логічні трюки дозволяють зробити саме рішення коротше, швидше і ефективніше. Це вже складніше, але чим більше методів може використовувати програміст, тим йому буде простіше оптимізувати програму і довести, що його рішення – краще.

Комбінаторна задача, яка найбільш часто зустрічається програмістами – вивести кількість елементів, які необхідно перебрати, щоб отримати рішення в залежності від деяких параметрів. Таким чином ви можете вивести асимптотичну складність алгоритму. Комбінаторні задачі сформульовані у вигляді завдання підрахунку кількості елементів деякої (в математиці використовують термін потужність) безлічі. Щоб вирішувати такі завдання, корисно мати базові знання в теорії множин з розряду властивостей операцій над ними. Тоді задача зводиться до вираження шуканої безлічі через безлічі, потужності яких обчислюються за відомими правилами. Для підрахунку кількості елементів застосовуються правила множення або додавання, числа сполучень або розміщень. Хоча є і більш складні завдання.

Також у програмуванні часто доводиться застосовувати імовірнісний підхід, для того щоб оцінити середню швидкість роботи алгоритму або ж підігнати параметри рішення задачі під ті запити, які найчастіше зустрічаються на практиці.

Щоб знайти екстремум (максимум або мінімум) функції, треба взяти її похідну і прирівняти до нуля. Рішення рівняння дає локальний екстремум. Але якщо потрібно шукати максимум не на проміжку, а тільки по цілих числах, то вже потрібно буде замислитися, яке з сусідніх цілих чисел потрібно вибрати. Коли завдання багатовимірне, варіантів з цілими числами стає все більше, і вибирати доводиться все складніше. Буває, що в таких завданнях можна знайти точне рішення тільки повним перебором, а іноді необхідно відмовитися від точного рішення, і використовувати деякі наближення. Тоді застосовують чисельні методи.

В даний час для оптимального прийняття рішень, в тому числі і в створенні ігор. а також для оптимізації витрат ресурсів комп'ютера, одним з апаратів математичної теорії, який найбільш частіше використовується є лінійне програмування.

Математикою в іграх може називатися просто додавання X і Y, маніпулювання різними формулами, тощо, однак в деяких випадках над реалізацією будь-якого правила в грі потрібно подумати не одну годину. Чим складніше будуть перетворення і дії, тим складніші математичні інструменти використовуються. Саме тому компанії, що займаються розробкою ігор, вимагають від своїх співробітників знання математики і алгоритмів. Знання таких речей не просто допоможе розробити логіку гри, але і якісно оптимізувати саму гру, знаходячи альтернативні шляхи, які допомагають уникнути зайвих обчислень.

Література:

1. Лінійна алгебра та аналітична геометрія: Навч. посібник / В. В. Булдигін, І.В. Алєксєєва, В.О. Гайдей, О.О. Диховичний, Н. Р. Коновалова, Л. Б. Федорова; за ред. проф. В.В. Булдигіна. – К.: ТВіМС, 2011. – 224 с.
2. Акулич И.Л. Математическое программирование в примерах и задачах./И.Л. Акулич. - Минск : Лань, 2011. – 352 c.
3. Чередникова А.В. Введение в теорию графов / А.В. Чередникова, И.В. Землякова. – Кострома: Издательство Костромского государственного технологического университета, 2011. – 24 с.
4. Physics for Game Developers, Second Edition / David M. Bourg, Bryan Bywalec. – USA: O’Reilly, 2013. – 551 c.
5. Practical Linear Algebra: A Geometry Toolbox / Gerald Farin, Dianne Hansford. – USA: CRC Press, 2013. – 514 c.