КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (26), 2024
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2024.26.626
УДК 519.6

Трофименко Олена Григорівна
к.т.н., доцент, доцент кафедри інформаційних технологій
Національний університет «Одеська юридична академія», Одеса, Україна
ORCID ID: 0000-0001-7626-0886
trofymenko@onua.edu.ua

Задерейко Олександр Владиславович
к.т.н., доцент, доцент кафедри інформаційних технологій
Національний університет «Одеська юридична академія», Одеса, Україна
ORCID ID: 0000-0003-0497-9861
zadereyko@onua.edu.ua

Баландіна Наталія Миколаївна
старший викладач кафедри кібербезпеки
Національний університет «Одеська юридична академія», Одеса, Україна
ORCID ID: 0000-0002-3121-4517
nataliabalandina2103@gmail.com

Толокнов Анатолій Арнольдович
асистент кафедри інформаційних технологій
Національний університет «Одеська юридична академія», Одеса, Україна
ORCID ID 0000-0002-2614-2109
arnoldovich1@gmail.com

Гусельніков Ілля Михайлович
студент
Національний університет «Одеська юридична академія», Одеса, Україна
ORCID ID: 0009-0000-6212-5755
airslim001@gmail.com

ВЕКТОРНА АЛГЕБРА ДЛЯ РОЗРОБКИ ІГОР

Анотація. Статтю присвячено висвітленню питань важливості математичних навичок для розробки різноманітного програмного забезпечення (ПЗ) загалом і розробки ігор (GameDev) зокрема. Адже саме за допомогою математичного апарата в програмному коді реалізується побудова й відображення ігрових сцен, поведінка, рух і взаємодія ігрових персонажів відповідно до подій, ігрового оточення та правил гри. В статті на конкретних практичних прикладах проаналізовано роль застосування векторної алгебри у розробці ігор з метою зацікавлення студентів ІТ-галузі у вивченні відповідних розділів вищої математики. Так, вектори в ігрових програмах часто використовують для опису фундаментальних властивостей ігрового персонажа: положення, швидкість руху, відстань між двома об'єктами тощо. Кожен об'єкт у грі має свої координати, які визначають його положення у віртуальному світі. Застосування векторної алгебри дозволяє програмістам точно визначити шлях, яким має рухатися персонаж чи то інший об'єкт, забезпечуючи при цьому плавність та реалістичність його руху. Розглянуто конкретні приклади фрагментів програмного коду мовою C# для ігрового середовища Unity 3D, які демонструють методи руху гравця у віртуальному просторі. Математичне моделювання поведінки гравця (ігрового об'єкта) у декартовій системі координат ігрового поля є неможливим без знання векторної алгебри та тригонометрії. Наведені приклади застосування елементів векторної алгебри є наочною демонстрацією актуальності та важливості математичних компетентностей для фахівців з GameDev. Математичні знання можуть допомогти розробникам ПЗ розробляти високоякісні програмні продукти. Тому на етапі навчання викладачам математики важливо зацікавити студентів ІТ-галузі у вивченні відповідних розділів вищої математики. Ефективним на цьому шляху є надання інформації щодо доцільності і можливого практичного застосування

© О.Г. Трофименко, О.В. Задерейко, Н.М. Баландіна, А.А. Толокнов, І.М. Гусельніков, 2024

відповідних математичних знань у сфері розробки ПЗ на конкретних прикладах з підкресленням впливу математики на ІТ-кар'єру.

Ключові слова: розробка ігор; векторна алгебра; Unity 3D; вектор; математичні навички; математика в ІТ; розробка програмного забезпечення.

ВСТУП

При навчанні на університетських освітніх програмах з підготовки програмістів подекуди як в колективі здобувачів вищої освіти, так і серед викладачів постають питання про доцільність вивчення тих чи інших розділів вищої математики з позиції подальшого практичного використання в професії.

Постановка проблеми. Нерідко серед студентів на молодших курсах навчання виникає нерозуміння ролі математики в майбутній професії і небажання вивчати відповідні математичні дисципліни. Проте більшість студентів вже з молодших курсів свідомо хочуть навчитися розробляти ігри і тому вибирають освітню траєкторію в напрямі GameDev.

З іншого боку, переважно на освітніх програмах ІТ-спеціальностей в українських вишах викладачі математичних дисциплін мають класичну університетську математичну освіту і доволі обмежене, зазвичай суто теоретичне розуміння можливого практичного застосування математичних компетентностей в розробці програмного забезпечення (ПЗ) загалом і GameDev зокрема, щоб пояснити практичні аспекти використання і тим самим зацікавити студентів у вивченні відповідних тем своїх дисциплін.

Аналіз останніх досліджень і публікацій. Роль і важливість вивчення математики для розробки різноманітного ПЗ досліджували різні науковці. Так, у дослідженні [1] дійшли висновку, що математика часто ігнорується в освітніх програмах з комп'ютерних наук, хоча вона має значний вплив на кар'єру ІТ-фахівців. З досвіду автора цього дослідження більшість студентів на ІТ-спеціальностях, особливо в перші роки навчання, не бачать переваг математики для своєї майбутньої кар'єри. А тому математика сприймається як абстрактний предмет, який не має жодного стосунку до роботи професійного розробника ПЗ. У статті [2] висвітлено різний вплив використання математики у різних областях інженерії програмного забезпечення: від критично важливого у сфері кібербезпеки та шифрування даних до менш вагомого у веброзробці та тестуванні ПЗ. Дослідники [3], [4] доводять, що програмування 3D-графіки та анімації в іграх потребує реалізації математичних рівнянь. У роботі [5] показано, як можна векторну алгебру застосовувати для керування рухами ігрових об'єктів. Автор роботи [6] пропонує інтегрувати математичні основи в міжнародний стандарт з інженерії програмного забезпечення IEEE Software Engineering Body of Knowledge (SWEBOK).

Мета статті: проаналізувати на прикладах роль застосування векторної алгебри у розробці ігор з метою зацікавлення студентів ІТ-галузі у вивченні відповідних розділів вищої математики.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Аналіз вакансій розробника ігор (Game Developer) на українському ринку IT-праці (https://www.work.ua/jobs-it-game+developer/) показує переважну відсутність чітких вимог до спеціальної математичної підготовки чи то до володіння якимось математичним професійним апаратом знань. Водночас популярними є професійні, навчальні курси з розробки ігор, де на прикладах вивчається математична складова для GameDev [7], [8]. Адже саме за допомогою математичного апарата в програмному коді реалізується побудова й відображення ігрових сцен, поведінка, рух і взаємодія ігрових персонажів відповідно до подій, ігрового оточення та правил гри.

Загалом гра розглядається як математична система, яка неперервно перемикається між підзадачами прийому вхідних даних від гравця, їх математично-програмним обробленням та надсиланням вихідних даних, переважно у графічній формі для візуалізації реакції гравця як результат змінення вхідних даних на певній ітерації гри. Важливу роль для реалізації динаміки гри відіграють знання з базової алгебри та тригонометрії з огляду математичного моделювання поведінки гравця (чи то ігрового об’єкта) у декартовій системі координат ігрового поля.

Зважаючи на суттєву роль математичних компетентностей для розробника ігор, важливо ще під час навчання на молодших курсах бакалаврату з комп’ютерних наук ефективно та результативно викладати математику студентам відповідних спеціальностей. Добре, коли викладач математичних дисциплін вибудовує і подає навчальний матеріал цікаво і захоплююче для студентів [9]. Гарною, переконливою стратегією для цього є підкреслення впливу математики на кар’єру, наприклад, в GameDev. Для кожної математичної теми варто надавати пояснення і конкретні приклади можливого практичного її використання в IT-сфері.

Досвідчені фахівці та викладачі математики [1] радять наводити паралелі зв’язків між математичною теорією та IT-професіями, що допомагають підтримувати зацікавленість аудиторії та бажання вчитися. Крім того, викладачеві математики корисно ділитися своїм особистим досвідом використання математики в IT-дослідженнях і професії. Такий підхід може реально зацікавити та захопити студентів, які можуть вирішити продовжити досліджувати відповідний математичний апарат у проєктній групі чи то науковому гуртку під керівництвом викладача. Обмін особистим професійним досвідом як ніщо інше надихає, мотивує і сприяє дискусіям та роздумам серед студентів про власні навички й інтереси, каталізує амбіції професійного розвитку.

Розглянемо приклади того, як одне з найбільш абстрактних і складних для студентів понять математики «вектори» можна пов’язати з практичним використанням в сфері GameDev. Наразі векторна алгебра як розділ математики є визначальним інструментом у розробці цікавих та захоплюючих ігор. Вектори відіграють важливу роль у визначенні руху ігрових об'єктів, взаємодії між ними та створенні реалістичного світу гри. Вектори є фундаментальною математичною концепцією, яка дозволяє описати напрямок і величину.

В ігрових програмах вектори часто використовують для опису фундаментальних властивостей ігрового персонажа: положення, швидкість руху, відстань між двома об’єктами тощо. Кожен об’єкт у грі має свої координати, які визначають його положення у віртуальному світі. Застосування векторної алгебри дозволяє програмістам точно визначити шлях, яким має рухатися персонаж чи то інший об'єкт, забезпечуючи при цьому плавність та реалістичність його руху [10].

ISSN 2663 - 4023

Приклад коду мовою C# для ігрового середовища Unity 3D демонструє метод руху гравця у двовимірному просторі, де метод `Update()` виконується циклічно з кожним оновленням кадра:

```
public float moveSpeed = 5f; // Швидкість руху гравця
void Update()
{
 float horizontalInput = Input.GetAxis("Horizontal");
 float verticalInput = Input.GetAxis("Vertical");
 Vector movement = new Vector(horizontalInput,
                 verticalInput, 0f) * moveSpeed *
                 Time.deltaTime;
 transform.pos += movement;
}
```

Тут `moveSpeed` — це швидкість руху гравця; `Time.deltaTime` — час між поточним та попереднім кадрами, його значення відповідає за те, щоб рух гравця був плавним, без ривків. Це значення підлаштовується автоматично під фактичну частоту кадрів монитора на конкретному комп'ютері.

Виклик конструктора `Vector` з параметрами `horizontalInput`, `verticalInput` та `0f` створює вектор `movement`, який задає напрям руху гравця у двовимірному просторі, відповідно до команд користувача (натискання клавіш керування рухом гравця). Вектор `movement` є нормалізованим, тому треба помножити його на швидкість гравця `moveSpeed` та на час оновлення одного кадру. По суті, такий вектор можна розглядати як кінцеву точку, в якій має опинитися гравець. Набір такого роду векторів на координатній площині формує відстані від початку координат (локації на ігровому полі) до гравця і до всіх наявних об'єктів на ігровому полі. Додавання вектора руху гравця `movement` до вектора попередньої позиції гравця (поточних координат об'єкта) `transform.pos` формує рух об'єкта.

Іншою важливою сферою застосування векторної алгебри і математики в цілому в розробці ігор є моделі штучного інтелекту для формування поведінки неігрових персонажів (Non-Player Character, NPC). Вектори можуть використовуватися для визначення маршруту руху NPC, їхньої реакції на зміни у середовищі гри та взаємодії з іншими об'єктами й персонажами [11]. Це дозволяє створювати глибокий ігровий світ, де кожен NPC має свою унікальну поведінку та реагує на дії гравця. Вектори використовуються для визначення маршрутів руху об'єктів у грі. Наприклад, вектори можуть вказувати напрямок руху, швидкість та точку призначення для NPC або об'єктів, які рухаються по заданому шляху. Так, в розробленій грі «Космічна дуель» векторну алгебру використано для реалізації поведінки ворожої ракети:

```
transform.pos = Vector.MoveTowards(transform.pos,
            player.transform.pos + addRandomToGo, speed *
            Time.deltaTime);
```

ISSN 2663 - 4023

Тут рух ракети відбувається шляхом періодичного додавання до координат поточної позиції ракети значень вектора `addRandomToGo`, який задає зміщення об'єкта в заданому діапазоні значень:

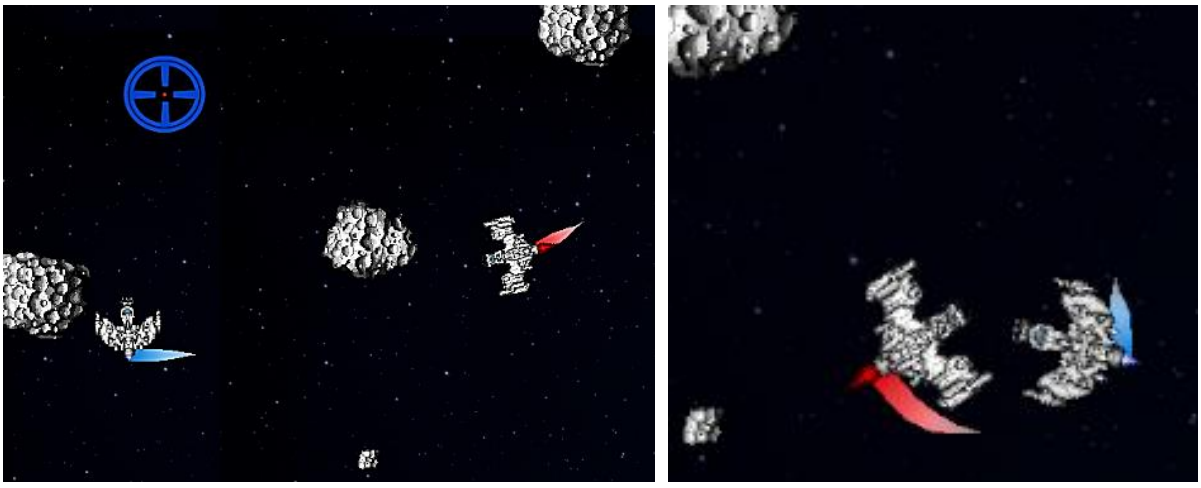
```
addRandomToGo = new Vector(Random.Range(-stopDistance+0.1f,
          stopDistance-0.1f), Random.Range(-stopDistance +
          0.1f, stopDistance - 0.1f));
```

Визначення відстані між поточною позицією об'єкта `(transform.pos)` і позицією гравця `(player.transform.pos)` у векторі визначається за формулою (1):

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

де d — відстань між двома точками (x_1, y_1) — координатами позиції гравця `(player.transform.pos)` та (x_2, y_2) — координатами позиції ворога `(transform.pos)`.

Випадковий параметр `addRandomToGo`, доданий до вектора позиції об'єкта `player.transform.pos` у методі `Vector.MoveTowards`, задає певну непередбачуваність руху transform.pos. Третій параметр `(speed * Time.deltaTime)` задає швидкість руху ворожого об'єкта (рис. 1). Блакитний та червоний сліди на цьому рисунку показують напрями руху гравця та ворога відповідно, оскільки в інший спосіб на статичному рисунку складно продемонструвати динаміку об'єктів у грі.



а *б*

Рис. 1. Візуалізація руху ігрових об'єктів:
а) ворожа ракета рухається вбік гравця;
б) рух ворога від гравця, коли відстань між ними мала

Рух об'єктів на ігровому полі може бути не лише прямолінійним. Для програмної реалізації повороту ігрового ворожого об'єкта потрібно обчислити кут між цим об'єктом і гравцем у двовимірній площині (рис. 2).

ISSN 2663 - 4023

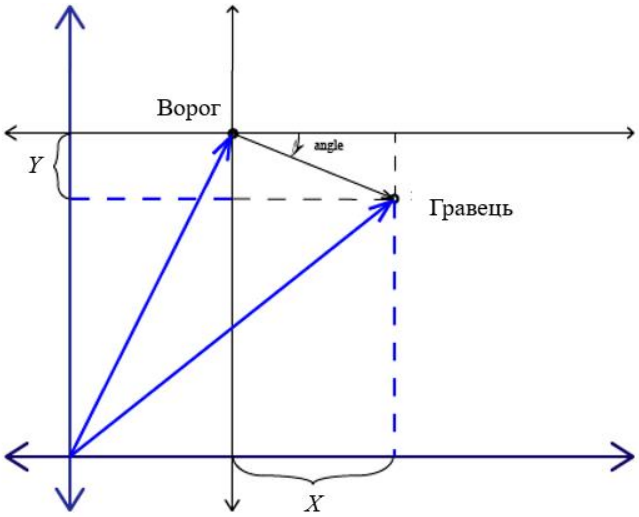


Рис. 2. Геометрія повороту ігрового об'єкта на заданий кут навколо певної осі

Тобто поворот формується через обчислення арктангенса кута між віссю *X* і координатами об'єкта (*x*, *y*) у декартовій системі координат як проєкції вектора на осі *X* та *Y* відповідно. Для цього у функцію `Mathf.Atan2()` передаються різниці координат по осі *Y* між гравцем і об'єктом та по осі *X* між гравцем і об'єктом як перший та другий аргументи відповідно:

```
float angle = Mathf.Atan2(playerPosition.y –
          transform.pos.y, playerPosition.x –
          transform.pos.x) * Mathf.Rad2Deg;
```

Тут функція `Mathf.Atan2()` обчислює значення кута в радіанах, а множення на константу `Mathf.Rad2Deg`, яка дорівнює 180/π, переводить це значення у градуси.

Фрагмент програмної реалізації повороту ігрового ворожого об'єкта вбік гравця у методі `Scale()`:

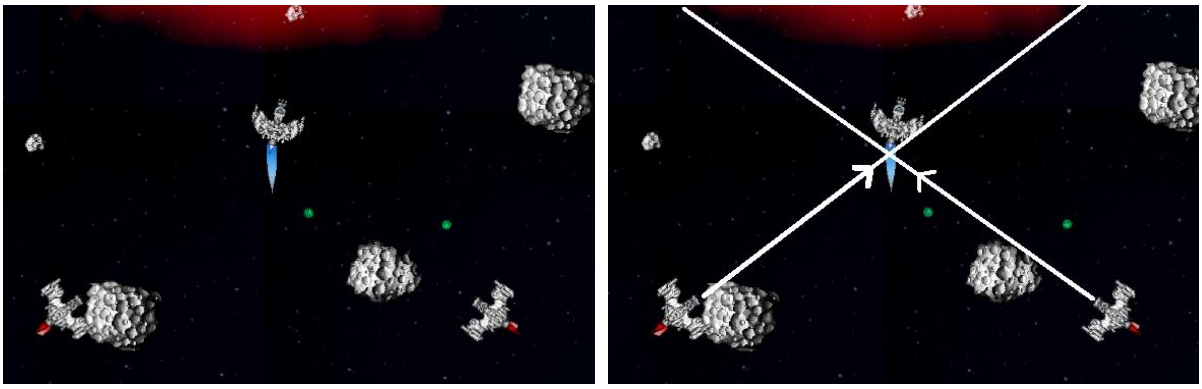
```
public void Scale()
     {
 Vector playerPosition = player.transform.pos;
      float angle = Mathf.Atan2(playerPosition.y –
                trasform.pos.y, playerPosition.x –
                transform.pos.x) * Mathf.Rad2Deg;
 Quaternion rotation = Quaternion.AngleAxis(angle,
Vector.forward);
      transform.rotation = Quaternion.Slerp(transform.rotation,
                rotation,
                5 * Time.deltaTime);
     }
```

Тут математична структура кватерніон `(Quaternion)` формує поворот на заданий кут навколо певної осі. Кут, на який слід повернути об'єкт, задається змінною `angle`, а вісь повороту — вектором `Vector.forward`, який вказує на вісь *Z* (0, 0, 1).

ISSN 2663 - 4023

Метод `Quaternion.AngleAxis` обчислює кватерніон, тобто поворот на `angle` градусів навколо осі Z.

Метод `Quaternion.Slerp()` здійснює плавну інтерполяцію між поточним обертанням об'єкта (`transform.rotation`) і цільовим обертанням (`rotation`) за допомогою сферичної лінійної інтерполяції (рис. 2, 3). Загалом сферична лінійна інтерполяція (Spherical Linear Interpolation, Slerp) є методом інтерполяції між двома кватерніонами, який забезпечує рівномірний і плавний перехід між ними [12]. Slerp використовується для обертання в дво- та тривимірному просторах. У наведеному прикладі забезпечення постійної швидкості обертання контролюється параметром `5 * Time.deltaTime`.



а *б*

Рис. 3. Поворот ворогів вбік гравця:
а) візуалізація поворотів двох ворожих об'єктів;
б) білими лініями показано результат сферичної лінійної інтерполяції; коли гравець рухається, ворогам потрібен час, щоб повернутися до нього

Наведені приклади застосування елементів векторної алгебри є наочною демонстрацією актуальності та важливості математичних компетентностей для фахівців з GameDev. Векторна алгебра використовується для моделювання фізичних явищ і забезпечення реалістичного сприйняття гравцями ігрового світу. По-перше, завдяки застосуванню векторної алгебри можна реалістично відтворити гравітацію. Так, за допомогою векторів можна визначити силу тяжіння, яка діє на об'єкт, та відповідно визначити його шлях руху. По-друге, векторна алгебра дозволяє моделювати силу тертя між поверхнями у грі, що дозволяє реалістично моделювати різні варіанти взаємодії гравця з наявними на полі об'єктами. Тобто можна в різний спосіб відтворювати рух об'єкта при взаємодії з різними типами поверхонь: відштовхуватись від стіни чи то іншої поверхні, змінювати рух від удару іншого гравця або вибуху з урахуванням маси та відстані до об'єкта впливу тощо.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Математика в цілому, як і векторна алгебра зокрема, відіграє важливу роль у створенні реалістичної та захоплюючої поведінки об'єктів у комп'ютерних іграх, забезпечуючи численні аспекти ігрової механіки й відтворення графіки. Вектори використовують для подання положення, напрямку та швидкості руху, впливу зовнішніх сил, наприклад, сил тяжіння і тертя, та багатьох інших фізичних величин у дво- та

тривимірному просторах. Математичні знання потенційно можуть допомогти розробникам ПЗ розробляти високоякісні програмні продукти.

Ступінь застосування математики залишається предметом активних дебатів між тими, хто виступає за сухий класичний підхід викладання усіх математичних дисциплін «по-максимуму», і тими, хто виступає за мінімізацію математичної складової у підготовці майбутніх програмістів. Насправді обсяги використання інженером-програмістом спеціальних математичних знань залежать від конкретної предметної області, в якій працює фахівець.

На етапі навчання викладачам математики важливо зацікавити студентів ІТ-галузі у вивченні відповідних розділів вищої математики. Адже з різних причин, у тому числі через складність сприйняття дітьми знань у дистанційній формі навчання, до якої українське суспільство перейшло спочатку через пандемію COVID-19, а згодом через війну, спостерігається зниження якості математичних знань. Тому важливими є пошуки можливих шляхів пробудити бажання вчитися. Ефективним на цьому шляху є надання інформації щодо доцільності та можливого практичного застосування відповідних математичних знань у сфері розробки ПЗ на конкретних прикладах з підкресленням впливу математики на ІТ-кар'єру.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Neri, F. (2021). Teaching Mathematics to Computer Scientists: Reflections and a CaseStudy. *SN Computer Science*, Springer, *2 (75)*. https://doi.org/10.1007/s42979-021-00461-7
2. O'Regan, G. (2023). Software Engineering Mathematics. In: *Mathematical Foundations of Software Engineering. Texts in Computer Science*. Springer, Cham. https://doi.org/10.1007/978-3-031-26212-8_2
3. Ghorashi, A., & Ghorashi, M. (2020). Theoretical and computational analysis of the falling ladder problem. *SN Comput Sci, 1(20),* 1–11. https://doi.org/10.1007/s42979-019-0019-7
4. Bing, L., Huiying, L., & Vinh Ph. (2022). 3D Animation Graphic Enhancing Process Effect Simulation Analysis. *Wireless Communications and Mobile Computing*, *9208495, 11*. https://doi.org/10.1155/2022/9208495
5. Sung, K., & Smith, G. (2023). Vectors. In: *Basic Math for Game Development with Unity 3D*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-9885-5_4
6. Gopal, T. V. (2022). Teaching Mathematics with the Software Engineering Body of Knowledge. *Innovative STEM Education*, *4*, 8–12. https://doi.org/10.55630/STEM.2022.0401
7. *Game Dev Math: Ultimate guide to polishing your game*! (n. d.). https://www.udemy.com/course/game-dev-math-ultimate-guide-to-polishing-your-game/
8. *Mathematics for Computer Games Development using Unity*. (n. d.). https://www.udemy.com/course/games_mathematics/
9. Трофименко, О. Г., Прокоп, Ю. В., Чепурна, О. Є., & Баландіна, Н. М. (2023). Роль математики у різних сферах розробки програмного забезпечення. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія «Технічні науки», 34(73), 4*, 117–123. https://doi.org/10.32782/2663-5941/2023.4/19
10. *Mathf. Unity documentation*. (n. d.). https://docs.unity3d.com/Manual/class-Mathf.html
11. Burzynski, D. (2023). Mathematics for Game Developers. *Teacher's edition*. Downey Unified School District, 163.
12. Young, K., Dat, H., Ashish, Sh., Wen-Kai. Ch., & Ser-Nam L. (2024). Spherical Linear Interpolation and Text-Anchoring for Zero-shot Composed Image Retrieval. *Computer Vision and Pattern Recognition*. 1–19. https://doi.org/10.48550/arXiv.2405.00571

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (26), 2024
ISSN 2663 - 4023

Olena Trofymenko
PhD, Associate Professor, Associate Professor of the Department of Information Technologies
National University "Odesa Law Academy", Odesa, Ukraine
ORCID ID: 0000-0001-7626-0886
trofymenko@onua.edu.ua

Olexander Zadereyko
PhD, Associate Professor, Associate Professor of the Department of Information Technologies
National University "Odessa Law Academy", Odesa, Ukraine
ORCID ID: 0000-0003-0497-9861
zadereyko@onua.edu.ua

Nataliia Balandina
Senior Lecturer of the Department of Cyber Security
National University "Odesa Law Academy", Odesa, Ukraine
ORCID ID: 0000-0002-3121-4517
nataliabalandina2103@gmail.com

Anatoly Tolocknov
Lecturer of the Department of Information Technologies
National University "Odesa Law Academy", Odesa, Ukraine
ORCID ID: 0000-0002-2614-2109
arnoldovich1@gmail.com

Illia Huselnikov
Student
National University "Odesa Law Academy", Odesa, Ukraine
ORCID ID: 0009-0000-6212-5755
airslim001@gmail.com

VECTOR ALGEBRA FOR GAMEDEV

Abstract. The article highlights the importance of mathematical skills development among IT students for working with various software and Game Development (GameDev) in particular. It is the mathematical tools in the software code that realize the game scenes construction and displaying as well as the game characters' behavior, movement and interaction correlated with events, game environment and rules. The objective of the article is to analyze the role of Vector Algebra application in the Game Development field using case studies to motivate IT students to take the appropriate sections of Higher Mathematics. In game programming vectors are often used to describe a game character's fundamental characteristics: position, velocity, and distance between two objects. Each object in the game has its own coordinates which determine its position in the virtual world. The implementation of Vector Algebra allows programmers to accurately determine a character's or object route ensuring the movement smoothness and realism. The specific examples of C# program code fragments for the Unity 3D game environment were considered to demonstrate the methods of the player's movement in the virtual space. The results demonstrate that the mathematical modeling of the player's or object behaviors on the playing field in the Cartesian coordinates is impossible without Vector Algebra and Trigonometry expertise. The given examples of Vector Algebra elements tailoring in curriculum are an evident demonstration of the relevance and importance of mathematical competences for Game Developers. Mathematical knowledge can be beneficial for Software Developers in designing high-quality products. IT students' motivation supported by Math teachers has an influential effect on studying the relevant sections of Higher Mathematics. The introductory information about the case studies, expediency, further application of relevant mathematical knowledge and its impact on IT career in Software Development field might be rather effective.

Keywords: Game Development; GameDev; Unity 3D; Vector Algebra; Vector; Math skills; Math in IT; Development Software.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Neri, F. (2021). Teaching Mathematics to Computer Scientists: Reflections and a CaseStudy. *SN Computer Science*, Springer, *2 (75)*. https://doi.org/10.1007/s42979-021-00461-7
2. O'Regan, G. (2023). Software Engineering Mathematics. In: *Mathematical Foundations of Software Engineering. Texts in Computer Science*. Springer, Cham. https://doi.org/10.1007/978-3-031-26212-8_2
3. Ghorashi, A., & Ghorashi, M. (2020). Theoretical and computational analysis of the falling ladder problem. *SN Comput Sci, 1(20)*, 1–11. https://doi.org/10.1007/s42979-019-0019-7
4. Bing, L., Huiying, L., & Vinh Ph. (2022). 3D Animation Graphic Enhancing Process Effect Simulation Analysis. *Wireless Communications and Mobile Computing*, *9208495*, *11*. https://doi.org/10.1155/2022/9208495
5. Sung, K., & Smith, G. (2023). Vectors. In: *Basic Math for Game Development with Unity 3D*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-9885-5_4
6. Gopal, T. V. (2022). Teaching Mathematics with the Software Engineering Body of Knowledge. *Innovative STEM Education*, *4*, 8–12. https://doi.org/10.55630/STEM.2022.0401
7. *Game Dev Math: Ultimate guide to polishing your game*! (n. d.). https://www.udemy.com/course/game-dev-math-ultimate-guide-to-polishing-your-game/
8. *Mathematics for Computer Games Development using Unity*. (n. d.). https://www.udemy.com/course/games_mathematics/
9. Trofymenko, O. H., Prokop, Yu. V., Chepurna, O. Ye., & Balandina, N. M. (2023). The role of mathematics in different areas of software development. *Scientific notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences, 34(73), 4*, 117–123. https://doi.org/10.32782/2663-5941/2023.4/19
10. *Mathf. Unity documentation*. (n. d.). https://docs.unity3d.com/Manual/class-Mathf.html
11. Burzynski, D. (2023). Mathematics for Game Developers. *Teacher's edition*. Downey Unified School District, 163.
12. Young, K., Dat, H., Ashish, Sh., Wen-Kai. Ch., & Ser-Nam L. (2024). Spherical Linear Interpolation and Text-Anchoring for Zero-shot Composed Image Retrieval. *Computer Vision and Pattern Recognition*. 1–19. https://doi.org/10.48550/arXiv.2405.00571

(cc) BY-NC-SA

This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.