УДК 51-78

РАЗДЕЛЫ МАТЕМАТИКИ, ПРИМЕНЯЕМЫЕ В РАЗРАБОТКЕ ВИДЕОИГР

Лариса Ивановна Никонорова
кандидат сельскохозяйственных наук, доцент
lenaniknrva@rambler.ru
Денис Владимирович Топильский
студент
denis.topiskiy@mail.ru
Мичуринский государственный аграрный университет
г. Мичуринск, Россия

Аннотация. В статье рассмотрены разделы математики, применяемые в игровой разработке, а также как математические концепции используются при разработке видеоигр.

Ключевые слова: видеоигры, математика, инкапсуляция, Unity 3D, язык программирования C#.

Математический аппарат представляет собой фундаментальную основу для построения алгоритмического ядра видеоигр. Сложные математические модели и методы, заимствованные из различных разделов математики, инкапсулированы в архитектуре игровых движков в виде наборов высокоуровневых программных абстракций — классов и функций. Данная инкапсуляция позволяет абстрагироваться от низкоуровневой реализации вычислений, перенося фокус разработчика на проектирование игровой логики и контента.

В качестве предмета исследования выбран игровой движок Unity 3D и язык программирования C#. Данный выбор обусловлен комплексом факторов: движок Unity обеспечивает оптимальный баланс между доступностью, функциональной гибкостью и производительностью. Язык C# предназначен для взаимодействия с API движка и написанием игровой логики.

Основной целью является систематизация ключевых разделов математики, применяемых в игровой разработке, с последующей демонстрацией их конкретной реализации через высокоуровневые интерфейсы (API) движка Unity 3D на языке C#. В результате проводится детальный разбор того, как математические концепции используются при разработке видеоигр.

Элементарная алгебра, оперирующая скалярными величинами, составляет основу для формализации игровых систем. Её аппарат обеспечивает алгоритмическую базу для расчёта параметров объектов, временных интервалов и нелинейных прогрессий. В Unity 3D этот слой реализован через базовые операторы C# и статический класс Mathf, предоставляющий оптимизированные математические функции. Ключевые методы включают:

- Mathf.Clamp(value, min, max) — ограничивает значение заданным диапазоном (min ≤ value ≤ max), обеспечивая нормирование параметров (например, уровня здоровья: health = Mathf.Clamp(health, 0, 100)).
- Mathf.Lerp(a, b, t) и Mathf.LerpUnclamped(a, b, t) — выполняют линейную интерполяцию (a + (b - a) * t) для плавных переходов между

состояниями, что применяется в процедурной анимации и динамическом изменении величин.

- Дополнительные функции (Mathf.Pow(), Mathf.Sqrt(), Mathf.Exp(), методы округления) расширяют инструментарий для решения нелинейных задач, моделирования физических зависимостей и дискретизации данных.

Булева алгебра, формализующая операции над двоичными переменными, реализуется в C# посредством логических операторов (&&, ||, !, ==, !=) и конструкций управления потоком. Условный оператор if-else обеспечивает ветвление программы на основе оценки предикатов, соответствующих логическим выражениям. Для выбора среди множества дискретных значений используется оптимизированный оператор switch. Высокоуровневый API Unity предоставляет поток булевых значений, выступающих входными данными для данных конструкций. Типичный пример — проверка нажатия на клавишу Space: if (Input. Get Key Down (KeyCode.Space)).

Центральным аппаратом для описания и манипуляций в игровом пространстве служит векторная математика. Классы Vector2 и Vector3 инкапсулируют не только координаты, но и предоставляют набор методов для операций над векторами. Ключевые высокоуровневые методы включают:

- Vector3.Cross() (векторное произведение). Основное применение — определение нормали к поверхности: Vector3 normal = Vector3.Cross(edge1, edge2).normalized;.
- Vector3.Distance() вычисляет евклидово расстояние между двумя точками.
- Vector3.Lerp(), Vector3.Slerp() и Vector3.MoveTowards() реализуют плавную интерполяцию координат: линейную, сферическую и равномерное движение. Пример: Vector3.Lerp(start, end, t);.

Свойство normalized возвращает единичный вектор направления (результат деления вектора на его длину), используемый для задания направления без влияния величины.

Для корректного представления и интерполяции трёхмерных вращений, исключая проблемы углов Эйлера (gimbal lock), применяются кватернионы. Класс Quaternion в Unity инкапсулирует соответствующие операции:

- Quaternion.Euler() конвертирует углы Эйлера в кватернион. Данный метод создания кватернионов используют чаще, чем создание кватерниона с помощью new Quaternion(), так как его проще использовать. Пример: Quaternionrotation = Quaternion.Euler(0,0,0);.
- Quaternion.Slerp() выполняет сферически-линейную интерполяцию вращений для их плавного изменения.Quaternion rotation = Quaternion.Slerp(start, end, t);. Применятся для плавного вращения.

Тригонометрические функции Sin(), Cos(), Tan(), Asin(), Acos(), Atan(), Atan2(), предоставляемые классом Mathf, служат инструментарием для параметризации и анализа угловых зависимостей, моделирования периодичности и колебаний, что является основой для решения геометрических задач.

Аппарат аналитической геометрии, изучающий геометрические объекты методами алгебры, реализован в Unity через систему физики и специализированные геометрические утилиты. Данные высокоуровневые API предоставляют инструментарий для решения задач на пересечение, определения взаимного расположения и пространственного анализа примитивов в реальном времени. Ключевые методы включают:

- Physics. Raycast() и Physics. RaycastAll() — реализуют алгоритмы испускания математического луча (заданного точкой и направлением) и определения его пересечения с коллайдерами в сцене. Используются для решения задач взаимодействия луча с геометрией.

Дискретная математика, оперирующая конечными, счётными множествами, составляет теоретическую основу для проектирования логики, искусственного интеллекта и структур данных в компьютерных играх. Её ключевые разделы находят прямое отражение в архитектуре игровых систем. В основном данная дисциплина используется для работы с моделью игрового Ии

Finite State Machine. Данная модель ИИ в математике это M = (Q, Σ, δ, q0, F), где Q — конечное множество состояний, служит доминирующей моделью для детерминированного ИИ. В Unity состояния кодируются перечислениями (enum), а функция переходов δ реализуется через ifelse или switch в методе Update().

Теория графов, изучающая свойства дискретных структур, состоящих из вершин и соединяющих их рёбер, находит системное применение в игровой разработке, главным образом в двух областях: пространственной навигации и проектировании логических систем.

- Навигация и поиск пути. Реализация навигации основана на преобразовании непрерывного игрового пространства в дискретный граф. В Unity для этого используется система NavMesh (Navigation Mesh), которая аппроксимирует проходимую геометрию сцены сеткой связанных выпуклых многоугольников, образующих граф. На этом графе компонент NavMeshAgent выполняет алгоритмы поиска кратчайшего пути (такие как A*), используя высокоуровневые методы, например, NavMesh.CalculatePath(). Визуальное построение и редактирование данного графа осуществляется компонентом NavMeshSurface.

Разделы математики и их применение в разработке игр приведены в таблице.

Раздел математики	Использование в разработке игр
Элементарная алгебра и арифметика	Базовые математические операции
Булева алгебра	Логическое ветвление
Векторная алгебра	Описания и манипуляции в игровом пространстве
Гиперкомплексные числа (кватернионы)	Плавный поворот 3D-модели
Тригонометрия	Вращение 2D-спрайта
Аналитическая геометрия	Проверка столкновения луча с объектом

Дискретная математика	Поиск пути
Теория графов	Алгоритм поиска пути A*, представленной как сетка

Проведённый анализ программных интерфейсов (API) движка Unity 3D позволяет сформулировать следующий вывод:

Математика, несомненно, остаётся фундаментальной основой для построения алгоритмического ядра видеоигр, обеспечивая формальные модели для описания пространства, физики, логики и поведения. Однако, ключевой особенностью современных игровых движков является глубокая инкапсуляция и абстракция сложных математических аппаратов. Эти аппараты представлены в виде высокоуровневых, семантически ясных классов и функций (API). Они предоставляются разработчику не в виде систем уравнений, а в виде интуитивных методов.

Таким образом, в контексте практической разработки игр, математика — это инструмент, предназначенный для непосредственного применения. Это позволяет разработчикам фокусироваться на создании контента, а не на реализации низкоуровневых математических алгоритмов.

Список литературы:

1. Документация методов Unity 3d // Unity – Scripting API – URL: https://docs.unity3d.com/ScriptReference/index.html

2. Руководство по C# — управляемый язык .NET // MicrosoftLearn – URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/

3. Разработка 3D-игр в Unity: руководство / Э. Дэвис, Т. Батист, Р. Крейг, Р. Станкел; перевод с английского П. М. Бомбаковой. Москва: ДМК Пресс, 2023. 298 с. ISBN 978-5-93700-254-9.

4. Данн Ф. Учебник по 3D математике для графики и разработке игр. 2021.

UDC 631.311

SECTIONS OF MATHEMATICS USED IN VIDEO GAME DEVELOPMENT

Larisa Iv. Nikonorova
candidate of agricultural sciences, associate professor
lenaniknrva@rambler.ru
Denis V. Topilsky
student
denis.topiskiy@mail.ru
Michurinsk State Agrarian University
Michurinsk, Russia

Annotation. The article discusses the branches of mathematics used in game development, as well as how mathematical concepts are used in the development of video games.

Keywords: video games, mathematics, encapsulation, Unity 3D, C# programming language.

Статья поступила в редакцию 25.02.2026; одобрена после рецензирования 20.03.2026; принята к публикации 31.03.2026.

The article was submitted 25.02.2026; approved after reviewing 20.03.2026; accepted for publication 31.03.2026.