УДК 51-74

МАТЕМАТИКА В РАЗРАБОТКЕ ПРИЛОЖЕНИЙ

Масюк Е.П., Таболич А.В.

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники» филиал «Минский радиотехнический колледж», г. Минск, Республика Беларусь

Научный руководитель: Тынкович В.В., преподаватель высшей категории дисциплин естественно-математического цикла

Аннотация. Исследована используемость математики в сфере «программирование». Приведены детально примеры использования математики в решении задач. Установлено, что математика используется во многих популярных приложениях. Рассмотрены формулы нахождения координат, теорема кодирования, как работает алгоритм создания плейлистов с любимой музыкой, алгоритмы работы жеста «зум» и создания ползунка с выбором времени.

Ключевые слова: программирование, математика, автоматизация, жесты, алгоритмы, кодирование, координаты

Введение. В нынешнее время оптимизировано множество процессов под жизнь человека. За нас выстраивается маршрут поездки, высчитывается предполагаемое время езды, создаются плейлисты с любимой музыкой. И самое интересное, так это то, что в каждом упомянутом процессе используется математика. Разработчикам для лучшего решения задачи нужно уметь анализировать и структурировать. Делить задачу на части, планировать и создавать алгоритмы решения. С этим как раз таки и поможет математика. Наша цель – рассказать, как уже используется математика в больших проектах и как бы мы ее использовали в решении задач. Мы решили разобрать области, где мы уже имеем опыт использования математики на реальных проектах.

Основная часть. Пожалуй, начнем с одного из самых популярных приложений. Приложение Яндекс Музыка позволяет слушать песни и альбомы.

Самой важной частью многих продуктов является система рекомендаций. Например, в ЯМ проще перечислить места, где не используются рекомендации, чем наоборот. Это очень важно, ведь метод «рандома» тут не подойдет – никто же не хочет услышать после Бетховена звук лютой бензопилы. Так как сервис построен на рекомендациях, то при отказе или сбоях такой системы он просто-напросто не может работать.

Как составляются рекомендации.

Обычно рекомендации строятся на том, что люди часто прослушивали треки A, B и C вместе: если очередной пользователь прослушивает записи A и B, то ему рекомендуют C. В случае с редкими песнями на помощь разработчикам приходит коллаборативная фильтрация. Она позволяет вычислять похожие записи на основе взаимодействий пользователей с контентом – даже если ни один человек не слушал их вместе.

Коллаборативная фильтрация работает так: нужно составить матрицу оценок пользователей, где по строкам расположатся пользователи, а по столбцам – треки. На пересечении строк и столбцов будут стоять пользовательские оценки: понравился им трек или нет (рисунок 1).

С подобными матрицами возникают две проблемы: во-первых, у них большая размерность – в «Яндекс.Музыке» можно прослушать сотни тысяч треков. Во-вторых, нам известен пользовательский фидбек только на части матрицы: люди слушают и оценивают небольшую долю мелодий – а по остальным данных нет. Для работы с такими разреженными матрицами разработчики используют сингулярное разложение или SVD.

С SVD позволяет сократить размерность матриц. Если R – матрица размера NxM и ранга r, то её можно разложить в произведение матрицы Mr и матрицы rM, сократив число параметров с (формула SVD 1)

$$N \times M$$

до (формула SVD 2)

$$(N + M) \times r$$

Рисунок 1 – Матрица SVD

Так как N и M на практике могут измеряться тысячами, а r обычно меньше десяти, это значительно упрощает работу с данными.

Обработав данные, алгоритм выдаёт список треков и исполнителей, которые могут понравиться пользователю. Считать его окончательной рекомендацией, однако, нельзя. Во-первых, список слишком длинный. Во-вторых, рекомендации должны быть разнообразными.

Окончательный список рекомендаций составляется с помощью «Матрикснет» – разработанного в «Яндексе» метода машинного обучения. «Матрикснет» обрабатывает список всех возможных рекомендаций – как полученных прогнозированием, так и составленных по другим источникам – и определяет, какие именно следует показать пользователю на главной странице «Яндекс.Музыки» и в каком порядке их расположить. Формула, по которой составляется лента рекомендаций, учитывает множество факторов – от сведений о том, сколько раз человек прослушал тот или иной трек, до времени суток: бывает так, что утром нравится одна музыка, а вечером – другая.

С недавним большим обновлением в сервисе появилась новая формула рекомендаций в “Моей волне” (рисунок 2), что позволило значительно увеличить разнообразие музыкального потока. Теперь пользователи на 30 % чаще слушают, а не просто встречают для себя новое, причем не только знакомых исполнителей, но и совсем новых – их в потоке стало больше на 23 %.

Рисунок 2 – Моя Волна в приложении Яндекс Музыки

Мы разобрались, как работают алгоритмы внутри приложения, теперь рассмотрим ситуацию, когда нам нужно при нажатии на кнопку “Show Modal” показать модальное окно и в этот же момент повернуть основной экран, тем самым сделать 3d эффект.

Данную задачу можно легко решить, если представить ее в пространстве как систему координат (рисунок 3):

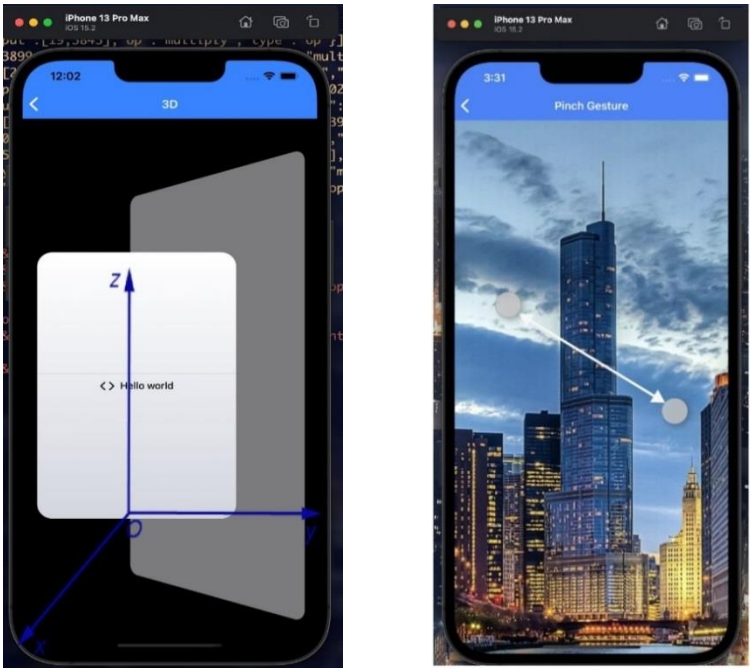

Рисунок 3 – Представление задачи в пространстве (слева), представление жеста «Zoom» (справа)

Все что нам остается сделать, так это задать свойство (в нашем случае свойство называется “perspective”), которое определяет расстояние между плоскостью z и пользователем для того, чтобы придать 3D-позиционируемому элементу эффект. Каждый трансформируемый элемент с $z > 0$ станет больше, с $z < 0$ соответственно меньше.

Далее мы рассмотрим пример реализации zoom в приложении Instagram:

Допустим, у нас есть экран с картинкой, и мы захотели ее приблизить. Для этого мы используем жест двумя пальцами, тем самым делая картинку ближе или дальше.

Теперь по порядку: нам предстоит отслеживать координаты пальцев. Обязательное условие - чтобы количество пальцев было два. В самом начале жеста мы должны записать значение координат, где начался жест по X и Y. Далее, когда пользователь начинает увеличивать/уменьшать изображение, мы должны сделать обработку этого события. Если же мы используем какую-либо библиотеку, то у нас будет коэффициент (назовем его zoom) значений текущих координат от начальных. Он нам как раз таки и поможет увеличивать само изображение. Если же такого коэффициента под капотом нету, тогда можно сравнивать сами координаты и получать тот же коэффициент. Осталось использовать получившийся коэффициент к нашему изображению (рисунок 3), (формула scale):

$$scale = scale \times zoom,$$

где scale является коэффициентом размеров самого изображения, а значение zoom может принимать как положительное, так и отрицательное число.

Пользователи активно взаимодействуют с сайтами: в интернет-магазинах они вводят данные своих банковских карт, на картографических сервисах прокладывают маршруты и измеряют расстояния, на музыкальных сайтах они транспонируют тональность песен и настраивают гитару по тюнеру, на сайтах конвертерах вводят сумму и сравнивают ее с другими валютами. И всё это должен кто-то запрограммировать.

Например, правильность номера банковской карты определяется по алгоритму Луна – это теория кодирования (рисунок 4).

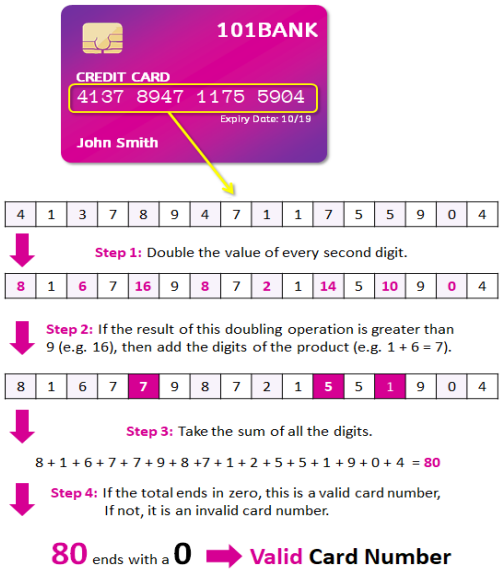

Рисунок 4 – Алгоритм Луна

Теперь давайте рассмотрим пример реализации выбора времени при помощи ползунка (рисунок 5):

Рисунок 5 – Пример дизайна приложения и представления координат

Со стороны жестов мы должны получать позицию пальца, чтобы изменять положение круга-ограничителя. Сперва мы должны конвертировать координаты в полярные координаты. Для этого мы должны узнать координаты относительно центра. Найти их можно найти по формулам, где c_x и c_y являются центром окружности:

$$x' = x - c_x$$

$$y' = -1 \times (y - c_y)$$

$$\theta = \tan^{-1}(\frac{y'}{x'})$$

$$r = \sqrt{x'^2 + y'^2}$$

Далее нам нужно получившиеся полярные координаты конвертировать в систему координат. Это делается для того, чтобы наша программа понимала canvas coordinates (двумерная сетка).

$$x' = r \times \cos(\theta)$$
$$y' = r \times \sin(\theta)$$

И осталось записать все в одну формулу.

Теперь мы можем узнать координаты наших кругов-ограничителей и посчитать, какой установлен интервал.

Заключение. Мы рассказали и показали, как используется математика в разработке приложений. Есть очень много случаев, когда нужна математика. На наш взгляд, мы показали одни из интересных примеров реализации математики в программировании. Тем самым, мы хотели донести, что математика играет не малую роль в разработке.

Список литературы

1.*Документация Яндекс, [Электронный ресурс]. Режим доступа: https://academy.yandex.ru/posts/kholodnye-polzovateli-i-mnogorukie-bandity Дата доступа : 29.03.2022 г.*

2.*Википедия, [Электронный ресурс]. Режим доступа: — https://en.wikipedia.org/wiki/Ellipse#Theorem_of_Apollonios_on_ conjjugate_diameters Дата доступа: 29.03.2022 г.*

UDC 51-74

MATHEMATICS IN SOFTWARE DEVELOPMENT

Masyuk E.P., Tabolich A.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus (style T-institution)

Tynkovich V.V., teacher of the highest category disciplines of the natural-mathematical cycle

Annotation. The use of mathematics in the field of "programming" is investigated. Examples of the use of mathematics in solving problems are given in detail. It has been established that mathematics is used in many popular applications. The formulas for finding coordinates, the coding theorem, how the algorithm for creating playlists with your favorite music works, the algorithms for the operation of the zoom gesture and the creation of a slider with a choice of time are considered.

Keywords. smart contracts, cryptography, encryption.