

Технология моделирования сплайнами позволила создать плавные контуры, свободно управлять структурой полигональной сетки для дальнейшего наложения текстур и облегчения процедуры рендеринга.

#### ЛИТЕРАТУРА

1. Компьютерная графика [Электронный ресурс] / Сайт «ВсеТренинги.ру. – Режим доступа: [https://vsetreningi.ru/schools/kompjuternaya\\_grafika/](https://vsetreningi.ru/schools/kompjuternaya_grafika/) – Дата доступа: 10.04.2021.

УДК 004.021

Студ. Т.С. Белявский

Науч. рук. доц. А.А. Дятко

(кафедра информатики и веб-дизайна, БГТУ)

### ВЫЧИСЛЕНИЕ ЗНАЧЕНИЯ ЧИСЛА $\pi$ МЕТОДОМ МОНТЕ-КАРЛО

Под методом Монте-Карло понимается численный метод решения математических задач при помощи моделирования случайных величин. Суть метода заключается в следующем: процесс описывается математической моделью с использованием генератора случайных величин, модель многократно обчисляется, на основе полученных данных вычисляются вероятностные характеристики рассматриваемого процесса. Например, чтобы узнать методом Монте-Карло, какое в среднем будет расстояние между двумя случайными точками в круге, нужно взять координаты большого числа случайных пар точек в границах заданной окружности, для каждой пары вычислить расстояние, а потом для них посчитать среднее арифметическое.

Метод имеет две основные особенности. Первая – простая структура вычислительного алгоритма. Вторая – ошибка вычислений, как правило, пропорциональна корню из отношения некоторой постоянной величины к числу испытаний [1].

Добиться высокой точности на таком пути невозможно. Поэтому обычно говорят, что метод Монте-Карло особенно эффективен при решении тех задач, в которых результат нужен с небольшой точностью. Следует отметить, что точность вычислений будет повышаться с увеличением количества испытаний.

**Формулировка задачи:** вычислить значение числа  $\pi$ , используя генератор случайных чисел, возвращающий числа от 0 до 1.

**Решение:** Для начала рассмотрим плоскость, в которой генерируются координаты точки (X, Y). Затем увеличим число генерируемых точек и добавим на плоскость окружность с радиусом 1 и цен-

тром в начале координат. Рассмотрим единичную окружность, представленную на плоскости. Так как площадь круга равна произведению квадрата радиуса на число  $\pi$ , мы можем утверждать, что площадь рассматриваемой окружности будет равна числу  $\pi$ . Площадь заданного квадрата будет равна 4, так как площадь квадрата равна квадрату длины одной из его сторон.

Можно заметить, что отношение площади квадрата к площади круга будет таким же, как и отношение количества точек, находящихся в квадрате к количеству точек, находящихся в круге. Следовательно,

$$Pi = (\text{точки в круге} / \text{точки в квадрате}) \cdot S_{\text{квадрата}}.$$

С помощью теоремы Пифагора мы можем узнать, находится ли точка в окружности или нет.

Программная реализация решения задачи (рисунок 1):

```
using System;

namespace Задача
{
    class Program
    {
        private static readonly Random Random = new Random();
        static void Main(string[] args)
        {
            int countCircle = 0;
            int countSquare = 0;

            Console.WriteLine("Введите количество точек");
            countSquare = Int32.Parse(Console.ReadLine()) ?? throw new InvalidOperationException();
            for (int i = 0; i < countSquare; i++)
            {
                double x = Random.NextDouble();
                double y = Random.NextDouble();

                if (Math.Sqrt(Math.Pow(x, 2) + Math.Pow(y, 2)) <= 1)
                {
                    countCircle++;
                }
            }

            double ret = (double)countCircle / countSquare;
            double Pi = ret * 4;
            Console.WriteLine("Число Pi равно : " + Pi);
        }
    }
}
```

Рисунок 1 – Программная реализация алгоритма

Результат решения задачи (рисунок 2–3).

```
Консоль отладки Microsoft Visual Studio
Введите количество точек
100000
Число Pi равно : 3,14376
```

Рисунок 2 – Выведение полученного значения  $Pi$  при 10 000 точек

```
Консоль отладки Microsoft Visual Studio
Введите количество точек
1700000
Число Pi равно : 3,1415129411764706
```

Рисунок 3 – Выведение полученного значения  $Pi$  при 1 700 000 точек

**Вывод:** метод довольно эффективен при решении задач, не требующих серьезной точности в результатах. Также не трудно видеть, что точность вычислений зависит от количества случайных величин, включенных в сумму. Причем, для увеличения точности вычислений

в 10 раз нужно увеличить в 100 раз. При решении некоторых задач для получения приемлемой точности оценки требуется брать очень большое число. А учитывая, что метод зачастую работает очень быстро, то реализовать последнее при современных вычислительных возможностях совсем не сложно.

#### ЛИТЕРАТУРА

1. Популярные лекции по математике 1968. Выпуск 46: Соболев, И.М. Метод Монте-Карло. – М.: Наука, 1968. – 64 с.

УДК 004.4'2

Студ. Т.В. Кулакова

Науч. рук. асс. Р.И. Белькевич

(кафедра информатики и веб-дизайна, БГТУ)

### ОБЗОР БЭКЕНД-ФРЕЙМВОРКОВ 2021 ГОДА

Согласно результатам опроса разработчиков, проведённого в 2020 году Stack Overflow, использование фреймворков играет огромную роль в IT-индустрии. В докладе речь идёт о трёх фреймворках – о Node.js/Express, Spring Boot и Django. Express, данный фреймворк создан на базе платформы Node.js. На базе данной платформы создано множество фреймворков, одним из популярных и является Express.

Spring Boot – это фреймворк для разработки бэкенд-приложений, основанный на Java, который, как и Node.js, используется для разработки микросервисов.

Django – это опенсорсный бэкенд-фреймворк, написанный на Python. Как известно, Python – это один из таких языков, которые пользуются наибольшей любовью разработчиков. И это – одна из основных причин того, что Django является одним из самых популярных серверных фреймворков.

Сравнение показано в таблице:

	<b>Node.js</b>	<b>Spring Boot</b>	<b>Django</b>
Плюсы	один и тот же код можно совместно использовать и на клиенте, и на сервере	если в процессе создания SpringBoot-приложения произошла ошибка, встроенный анализатор ошибок поможет справиться с проблемой	поддерживает паттерн проектирования MVC
Минусы	сильная зависимость от пакетного менеджера NPM	большая тяжеловесность и медлительность приложения	развивается медленно

Из обзора видно, что основной проблемой является медлительность процессов обработки данных.