УДК 004.421.5
DOI: 10.17586/0021-3454-2024-67-4-338-344

ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ЧИСЕЛ ДЛЯ ИНТЕРАКТИВНЫХ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ВНЕШНИХ ДАТЧИКОВ

Д. А. Булгаков

Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, Россия
dmbulg@gmail.com

Аннотация. Представлен метод получения псевдослучайных чисел для их дальнейшего использования при разработке интерактивных приложений на движке Unity со сбором информации от датчиков давления и цвета, подключаемых к микроконтроллеру Arduino. Метод предполагает использование результатов периодических измерений давления, температуры, освещенности и цветов по каналам RGB в помещении, их побитовый сдвиг на случайное число разрядов, получение „зерна“ генератора псевдослучайных чисел путем взятия остатка от деления после сравнения числа с текущим UNIX-временем. Разработано приложение, реализующее предложенный метод генерации псевдослучайных чисел. Показаны результаты тестирования генератора псевдослучайных чисел. Проведена проверка равномерности распределения и оценка коэффициента корреляции на выборке случайных чисел.

Ключевые слова: *случайные числа, генератор псевдослучайных чисел, интерактивное приложение, датчик давления, датчик цвета, Arduino, Unity*

Ссылка для цитирования: *Булгаков Д. А.* Генерация случайных чисел для интерактивных приложений с использованием внешних датчиков // Изв. вузов. Приборостроение. 2024. Т. 67, № 4. С. 338—344. DOI: 10.17586/0021-3454-2024-67-4-338-344.

RANDOM NUMBER GENERATION FOR INTERACTIVE APPLICATIONS USING EXTERNAL SENSORS

D. A. Bulgakov

St. Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russia
dmbulg@gmail.com

Abstract. A method is presented for obtaining pseudo-random numbers to be used further in the development of interactive applications on the Unity engine with the collection of information from pressure and color sensors connected to the Arduino microcontroller. The method involves using the results of periodic measurements of pressure, temperature, illumination, and colors on RGB channels in a room, bit shifting them by a random number of digits, obtaining the “grain” of a pseudo-random number generator by taking the remainder after comparing the number with the current UNIX time. An application is been developed that implements the proposed method of generating pseudorandom numbers. The uniformity of distribution is checked and the correlation coefficient is assessed using a sample of random numbers.

Keywords: random numbers, pseudo-random numbers generator, interactive applications, pressure sensor, color sensor, Arduino, Unity

For citation: Bulgakov D. A. Random number generation for interactive applications using external sensors. *Journal of Instrument Engineering.* 2024. Vol. 67, N 4. P. 338—344 (in Russian). DOI: 10.17586/0021-3454-2024-67-4-338-344.

Введение. Решение таких задач, как генерация одноразовых паролей и хэш-сумм, определение победителя в различных конкурсах и лотереях, генерация территорий в многопользовательских играх и метавселенных, невозможно представить без надежного генератора случайных чисел [1].

В отличие от физических процессов в природе, математические операции, выполняемые на компьютере, являются детерминированными, поэтому добиться истинной случайности от сгенерированных чисто алгоритмическими методами чисел невозможно. Сгенерированные на компьютере числа принято называть псевдослучайными, а генератор псевдослучайных чисел обозначать как ГПСЧ [2].

Использовать обычные арифметические операции для генерации псевдослучайных чисел, в частности в методе середины квадрата, впервые предложил Джон фон Нейман [3]. Конечно, метод не дает истинно случайных чисел, но доказательство того, что алгоритмические методы не могут предоставить истинную случайность результатов, наглядно подтвердил Дональд Кнут после реализации на компьютере сложного многошагового алгоритма „Алгоритм К“, который довольно быстро сошелся к одному 10-значному числу [4].

Алгоритмы ГПСЧ совершенствовались с течением времени. На сегодняшний день существует не один десяток достаточно надежных ГПСЧ, наиболее известны: Вихрь Мерсенна, используемый в функции RAND языка программирования Python, субтрактивный метод Кнута, используемый в методе Random языка C#, и алгоритм Fortuna, используемый в продуктах корпорации Apple. Тем не менее, любой ГПСЧ с ограниченными ресурсами рано или поздно зацикливается — начинает повторять одну и ту же последовательность чисел [5—7]. Длина циклов ГПСЧ в среднем составляет $2^{n/2}$, где n — объем памяти, который занимает внутреннее состояние алгоритма.

Для создания ГПСЧ, который удовлетворял бы равномерному закону распределения, обладал сходимостью моментов и позволял получать длительный период случайной последовательности, применения только алгоритма генерации недостаточно — в уравнения требуется включать числовые данные, полученные от различных недетерминированных событий, происходящих в окружающем мире [8].

Далее представлен вариант генератора псевдослучайных чисел на основе внешних данных, который может быть использован при разработке любых интерактивных приложений на игровом движке.

Описание предлагаемого решения. Для практических задач, не предъявляющих критических требований к безопасности, можно ограничиться измерением различных физических величин из окружающей среды. Эти данные затем будут преобразовываться в целые числа и использоваться в качестве параметров алгоритма ГПСЧ в программе, написанной на языке C#. Для демонстрации работы предлагаемого метода был взят микроконтроллер Arduino Uno R3. Схематичное изображение платы Arduino Uno с подписанными контактами приведено на рис. 1, к ней подключались [9—11]:

1) датчик атмосферного давления и температуры BMP280;

2) датчик цвета TCS3472.

В среде разработки Arduino IDE реализовано приложение „bmp280_plus_tcs3472.ino“ для инициализации и настройки датчиков, а на игровом движке Unity создано тестовое приложение „BMPTCSRNG“ — для считывания данных с датчиков.

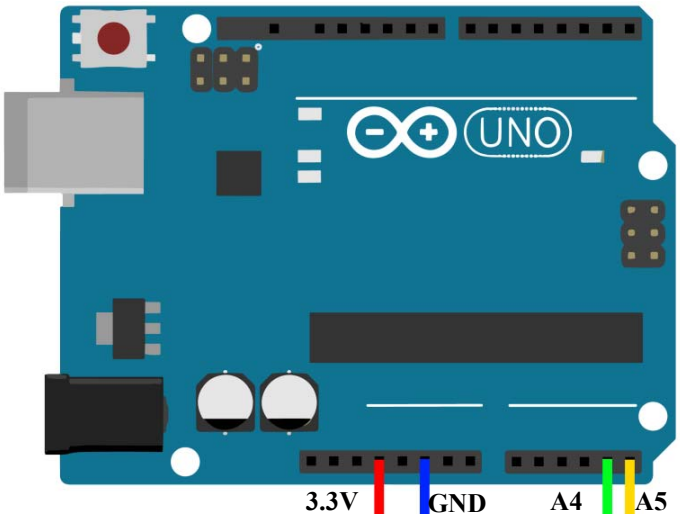

Рис. 1

Модуль измерения атмосферного давления и температуры построен на базе чипа BOSH BMP280 и заранее откалиброван на заводе (табл. 1). Модуль предоставляет пользователю два последовательных интерфейса обмена данными: SPI и I2C. К плате ArduinoUno датчик BMP280 подключается следующим образом: VCC—3.3V; GND—GND; SCL—A5; SDA—A4.

Таблица 1

Назначение контактов датчика BMP280

Контакт	Назначение
VCC	Питание 3,3 В
GND	Обеспечение нулевого потенциала (Земля)
SCK (SCL)	Ввод частоты по интерфейсам I2C и SPI
SDI (SDA)	Ввод данных по интерфейсам I2C и SPI
CSB	Выбор активного интерфейса I2C
SDO	Изменение адреса I2C

Датчик TCS3472 способен выводить данные о цвете поверхности по каналам R, G, B, C, а также интенсивности света, измеряемой в люксах (табл. 2). Датчик использует интерфейс I2C для передачи данных. К плате ArduinoUno датчик TCS3472 подключается следующим образом: VIN—3.3V; GND—GND; SCL—A5; SDA—A4.

Таблица 2

Назначение контактов датчика TCS34725

Контакт	Назначение
VIN	Питание 3,3/5 В
GND	Обеспечение нулевого потенциала (Земля)
3V3	Вывод напряжения 3,3 В
SCL	Ввод частоты по интерфейсу I2C
SDA	Ввод данных по интерфейсу I2C
INT	Выход прерывания, активный низкий уровень
LED	Вход управления светодиодами

Для получения показаний с датчиков требуется подключение библиотек „Adafruit_BMP280.h“ и „Adafruit_TCS34725.h“ с открытым исходным кодом.

Ниже приведен фрагмент кода скрипта обработки показаний с датчиков — метод GetTemperatureInfo(), считывающего показания температуры:

```
public void GetTemperatureInfo()
{
float number = FindNumberInString(outputLines[0]);
temperatureValue.text = number.ToString();
}
```

Значение температуры записывается в текстовое поле — элемент пользовательского интерфейса, который отображается на экране. Поскольку строка содержит не только числа, но

также текст и символы, то применяется метод-парсер FindNumberInString, удаляющий из строки все буквы и знаки. Вид интерфейса приложения с полученными данными приведен на рис. 2.

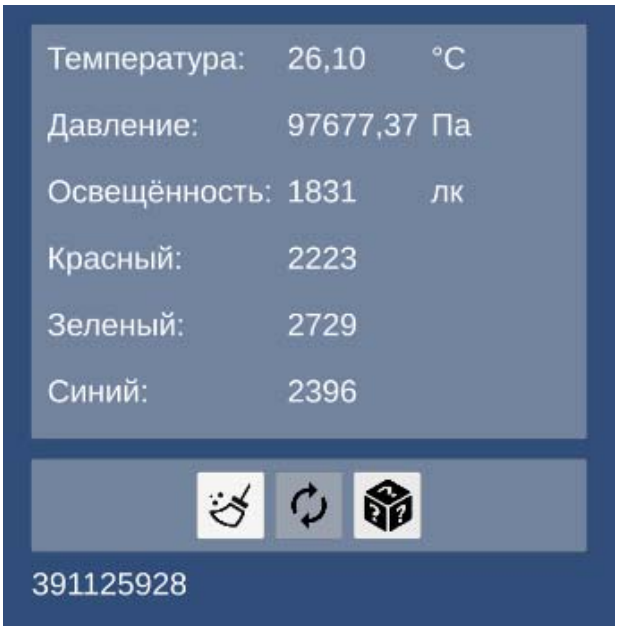

Рис. 2

Получение новых данных сопровождается выполнением следующей последовательности действий: закрытие последовательного порта, обнуление списка данных и открытие порта. Эти операции выполняются в методе ClearData.

Предлагаемый метод генерации псевдослучайных чисел включает следующие шаги:

1) сформировать массив числовых значений $a_i, i = \overline{1,n}$ на основе показаний датчиков, $n = 6$;

2) выполнить побитовый сдвиг влево значений $a_i, i = \overline{1,n}$ на r разрядов. Значение r разыгрывается методом Random.Next() — выполняется генерация псевдослучайного целого числа в диапазоне типа Integer;

3) выбрать одно из значений $a_i, i = \overline{1,n}$ методом Random.Next();

4) зафиксировать метку текущего времени $m(t)$ — UNIX-время;

5) если $m(f) > a_i$ и $a_i \neq 0$, то вычислить $A = m(f) \bmod a_i$, иначе $A = a_i \bmod m(f)$;

6) считать полученное значение A итоговым „зерном“ ГПСЧ;

7) сгенерировать на основании итогового „зерна“ псевдослучайное число, используя метод Random.Next (*d*), где *d* — правая граница диапазона.

Генерация случайного числа в приложении запускается нажатием на кнопку „Сгенерировать“ (см. рис. 2). Итоговое 32-битное число типа int выводится в текстовое поле в нижней части экрана.

Тестирование ГПСЧ. Как известно, ГПСЧ должны удовлетворять следующим требованиям [12, 13]:

— числа в генерируемой последовательности должны быть равномерно распределены и независимы, что проверяется статистическими тестами;

— период последовательности должен иметь возможно большую длину.

Равномерность распределения по выборке $A_1, A_2, ..., A_n$ проверяется путем определения эмпирических вероятностных характеристик — моментов и параметров распределения, а также их сравнения с теоретическими характеристиками распределения.

Для проверки равномерности распределения применялся частотный тест, последовательность его проведения следующая [14].

1. Получить выборку ГПСЧ: $A_1, A_2, ..., A_n$.

2. Разбить интервал ГПСЧ $[0, d]$ на K равных отрезков.

3. Подсчитать, сколько чисел попало в каждый из K отрезков, т.е. найти число попаданий $n_1, n_2, ..., n_k$.

4. Найти относительную частоту попадания в отрезки:

$$\hat{p}_1 = \frac{n_1}{n}, \hat{p}_2 = \frac{n_2}{n}, ..., \hat{p}_K = \frac{n_k}{n}.$$

5. Построить гистограмму частот $\hat{p}_1, \hat{p}_2, ..., \hat{p}_K$.

6. Оценить по полученной гистограмме сходимость каждой частоты $\hat{p}_i$ к вероятности $p = \frac{1}{K}$.

Согласно закону больших чисел, значения $\hat{p}_1, \hat{p}_2, ..., \hat{p}_K$ должны сходиться к p при $n \to \infty$.

Гистограмма частотного теста на выборке из 16 000 значений предложенного ГПСЧ приведена на рис. 3.

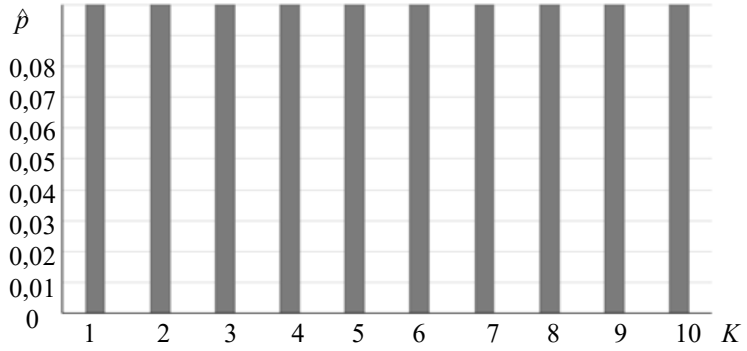

Рис. 3

Проверка статистической независимости ГПСЧ выполнена с помощью оценки линейной корреляции между величинами A_i и A_{i+s}, при $s \geq 1$, где s — шаг, с которым выбираются случайные числа [15—17].

На рис. 4 приведен график зависимости коэффициента корреляции R от длины случайной последовательности, из которого видно, что $R \to 0$.

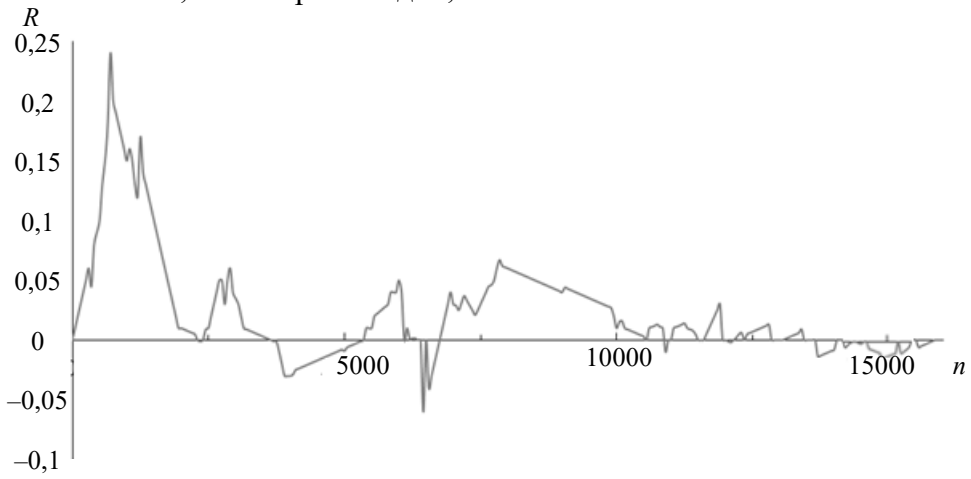

Рис. 4

Заключение. Предложенный метод генерации псевдослучайных чисел позволяет получать и применять разнородные данные из внешних источников с помощью известных методов в интерактивных приложениях, построенных на игровом движке Unity.

Метод предполагает использование результатов периодических измерений давления, температуры, освещенности и цветов по каналам RGB в помещении и получении „зерна“ генератора псевдослучайных чисел после применения определенных операций над измерениями.

Предложенный метод позволяет создать генератор псевдослучайных чисел, близкий по своим характеристикам к истинному источнику энтропии [18], что показало тестирование на равномерное распределение и отсутствие корреляции между случайными числами, выдаваемых генератором.

СПИСОК ЛИТЕРАТУРЫ

1. *Андреева Е. Г., Молчалин В. А.* Генератор псевдослучайных чисел в игровых механиках // Россия молодая: передовые технологии — в промышленность. 2023. № 1. С. 3—9. DOI 10.25206/2310-4597-2023-1-3-9. EDN GIMLYI.
2. *Чайко В. И.* Накопление случайности в генераторах псевдослучайных чисел // Исследования молодых ученых: Матер. XXXII Междунар. науч. конф. Казань, 20—23 февраля 2022 г. Казань: Молодой ученый, 2022. С. 10—15. EDN IPCVOY.
3. *Von Neumann J.* Various techniques use dinconnection with random digits // National Bureau of Standards Applied Mathematics Series. 1951. N 12. P. 36—38.
4. *Кнут Д. Э.* Искусство программирования. Т. 2. Получисленные алгоритмы. 3-е изд. М.: Диалектика, 2020. 832 с.
5. *Белов А. А., Калиткин Н. Н., Тинтул М. А.* Ненадежность известных генераторов псевдослучайных чисел // Журнал вычислительной математики и математической физики. 2020. Т. 60, № 11. С. 1807—1814. DOI 10.31857/S0044466920110046. EDN CTJCWS.
6. *Орлов М. А.*, *Нечаев К. А., Иванов Н. А.* Проблемы криптостойкости в современных ГПСЧ // Наука и бизнес: пути развития. 2022. № 4(130). С. 53—58. EDN SMUPYE.
7. *Романков С. В.* Методы генерации псевдослучайных чисел // Молодой ученый. 2022. № 33(428). С. 4—10. EDN EHKDWM.
8. *Dhirendra K., Chaurasia U., Mishra S.* Design of True Random Number Generator Using Fingerprint as an Entropy Source and Its Implementation in S-Box // J. of Circuits, Systems and Computers. 2021. Vol. 30, N 15. Art. no 2150285.
9. Arduino Software. Официальная документация и спецификации модели UnoR3 [Электронный ресурс]:<https://docs.arduino.cc/hardware/uno-rev3>. (дата обращения: 21.12.2023).
10. 3DiY (Тридиай). Датчик атмосферного давления BMP280 [Электронный ресурс]:<https://3d-diy.ru/wiki/arduino-datchiki/sensor-bmp280/>. (дата обращения: 21.12.2023).
11. Wave share Electronics. TCS34725 Color Sensor User Manual [Электронный ресурс]: <https://www.waveshare.com/w/upload/b/bb/TCS34725_Color_Sensor_user_manual_en.pdf>. (дата обращения: 21.12.2023).
12. *Дроздова И. И., Жилин В. В.* Генераторы случайных и псевдослучайных чисел // Технические науки в России и за рубежом: Матер. VII Междунар. науч. конф. М., 2017. С. 13—16.
13. *Гончарук В. С., Атаманов Ю. С., Гордеев С. Н.* Методы генерации случайных чисел // Молодой ученый. 2017. № 8(142). С. 20—23.
14. *Кутузов О. И., Татарникова Т. М.* Из практики применения метода Монте-Карло // Заводская лаборатория. Диагностика материалов. 2017. Т. 83, № 3. С. 65—70.
15. *Колесова Н. А.* Оценка качества генераторов последовательностей случайных чисел // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2011. № 1. С. 119—123.
16. *Григорьев А. Ю.* Методы тестирования генераторов случайных и псевдослучайных последовательностей // Ученые записки УлГУ. 2017. № 1. С. 22—28.

17. *Пахомов В. А., Титовская Е. П.* Исследование надежности генератора псевдослучайных последовательностей // Юный ученый. 2020. № 4(34). С. 70—75.

18. *Зубков А. М.* Энтропия как характеристика качества случайных последовательностей // Математические вопросы криптографии. 2021. Т. 12, № 3. С. 31—48. DOI 10.4213/mvk374. EDN RJVEOY.

Сведения об авторе

Дмитрий Алексеевич Булгаков — Санкт-Петербургский государственный университет аэрокосмического приборостроения, кафедра прикладной информатики (Кафедра 41); старший преподаватель; E-mail: dmbulg@gmail.com

Поступила в редакцию 04.12.23; одобрена после рецензирования 11.12.23; принята к публикации 08.02.24.

REFERENCES

1. Andreeva E.G., Molchalin V.A. *Young Russia: advanced technologies into industry*, 2023, no. 1, pp. 3–9, DOI 10.25206/2310-4597-2023-1-3-9. (in Russ.)
2. Chayko V.I. *Issledovaniya molodykh uchenykh* (Research by Young Scientists), Proc. of the XXXII Intern. Scientific Conf., February 20–23, 2022, Kazan, 2022, pp. 10–15. (in Russ.)
3. Von Neumann J. *National Bureau of Standards Applied Mathematics Series*, 1951*, no.* 12*, pp.* 36–38.
4. Knuth D. *The Art of Computer Programming, vol. 2: Seminumerical Algorithms*, Massachusetts, Addison-Wesley, 1997, 762 p.
5. Belov A.A., Tintul M.A., Kalitkin N.N. *Computational Mathematics and Mathematical Physics*, 2020, no. 11(60), pp. 1747–1753, DOI 10.31857/S0044466920110046.
6. Orlov M.A., Nechaev K.A., Ivanov N.A. *Science and business: ways of development*, 2022, no. 4(130), pp. 53–58. (in Russ.)
7. Romankov S.V. *Young scientist*, 2022, no. 33(428), pp. 4–10. (in Russ.)
8. Dhirendra K., Chaurasia U., Mishra S. *Journal of Circuits, Systems and Computers*, 2021, no. 15(30), pp. 2150285.
9. *Arduino Software*, https://docs.arduino.cc/hardware/uno-rev3.
10. https://3d-diy.ru/wiki/arduino-datchiki/sensor-bmp280/. (in Russ.)
11. *Wave share Electronics. TCS34725 Color Sensor User Manual*, https://www.waveshare.com/w/upload/b/bb/TCS34725_Color_Sensor_user_manual_en.pdf.
12. Drozdova I.I., Zhilin V.V. *Tekhnicheskiye nauki v Rossii i za rubezhom* (Technical Sciences in Russia and Abroad), Materials of the VII Intern. Scientific Conf., Moscow, 2017, pp. 13–16. (in Russ.)
13. Goncharuk V.S., Atamanov Yu.S., Gordeev S.N. *Young scientist*, 2017, no. 8(142), pp. 20–23. (in Russ.)
14. Kutuzov O.I., Tatarnikova T.M. *Factory laboratory. Diagnostics of materials*, 2017, no. 3(83), pp. 65–70. (in Russ.)
15. Kolesova N.A. *Vestnik of Astrakhan State Technical University. Series: Management, Computer Sciences and Informatics*, 2011, no. 1, pp. 119–123. (in Russ.)
16. Grigoriev A.Yu. *Scientific Notes of ULSU. Series: Mathematics and Information Technology*, 2017, no. 1, pp. 22–28. (in Russ.)
17. Pakhomov V.A., Titovskaya E.P. *Young scientist*, 2020, no. 4(34), pp. 70–75. (in Russ.)
18. Zubkov A.M. *Mathematical Aspects of Cryptography*, 2021, no. 3(12), pp. 31–48, DOI 10.4213/mvk374. (in Russ.)

Data on author

Dmitriy A. Bulgakov — St. Petersburg State University of Aerospace Instrumentation Department of Applied Informatics (Department 41); Senior Lecturer; E-mail: dmbulg@gmail.com

Received 04.12.23; approved after reviewing 11.12.23; accepted for publication 08.02.24.