



## ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА РЕЙНОЛЬДСА

**В. Д. Забурдин, Р. Р. Абдрашитов, А. В. Шахомиров**

Санкт-Петербургский государственный университет аэрокосмического приборостроения

*В статье рассматривался алгоритм имитации поведения стаи животных, птиц, рыб, насекомых, разработанный Крейгом Рейнольдсом. Этот алгоритм по сей день используется для симуляции коллективного движения и взаимодействия множества объектов в компьютерной графике, искусственном интеллекте и виртуальной реальности. Также была представлена реализация математической модели, улучшенной Джереми Г. Бондом, реализованной на языке C# с применением графической библиотеки Unity 2017.*

*Ключевые слова: компьютерная графика, моделирование, алгоритм Рейнольдса, боиды, рои.*

### **Для цитирования:**

*Забурдин, В. Д. Программная реализация алгоритма Рейнольдса / В. Д. Забурдин, Р. Р. Абдрашитов, А. В. Шахомиров // Системный анализ и логистика. – 2023. – № 3(37). – с. 21–26. DOI: 10.31799/2077-5687-2023-3-21-26.*

## SOFTWARE IMPLEMENTATION OF THE REYNOLDS ALGORITHM

**V. D. Zaburdin, R. R. Abdrashitov, A. V. Shakhomirov**

St. Petersburg State University of Aerospace Instrumentation

*The article discussed an algorithm for simulating the behavior of a flock of animals, birds, fish, and insects developed by Craig Reynolds. This algorithm is used to this day to simulate the collective movement and interaction of multiple objects in computer graphics, artificial intelligence, and virtual reality. An implementation of the mathematical model improved by Jeremy G. Bond, implemented in C# using the Unity 2017 graphics library, was also presented.*

*Keywords: computer graphics, modeling, Reynolds algorithm, boids, swarms.*

### **For citation:**

*Zaburdin, V. D. Software implementation of the Reynolds algorithm / V. D. Zaburdin, R. R. Abdrashitov, A. V. Shakhomirov // System analysis and logistics. – 2023. – № 3(37). – p. 21–26. DOI: 10.31799/2077-5687-2023-3-21-26.*

### **Введение**

Долгие годы после зарождения программирования, исходные коды представляли собой огромный объем текста, который старался уследить за всем сразу. Хотя такой “монолитный” подход и имеет право на существование, куда более разумным и действенным методом при написании масштабных проектов будет использование модульного подхода с основными принципами объектно-ориентированного программирования. Хотя сейчас почти каждому программисту и очевидны преимущества ООП перед бесконечными строками кода, в моменты его зарождения многие оставались неуверенными в данном подходе. Одним из людей, вложивших свой вклад в признание модульного и объектно-ориентированное программирования, был Крейг Рейнольдс разработавший принцип поведения объектов, которым позже он дал имя боиды.

### **Основные принципы боидов**

В 1987 году в журнале “Компьютерная графика” публикуется статья Крейга Рейнольдса под названием “Стаи, стада и школы: Распределенная поведенческая модель” [1]. В этой статье автор однозначно определяет метод реализации поставленной задачи.

Этот подход предполагает, что стая – это просто результат взаимодействия поведения отдельных птиц. Чтобы моделировать стаю, мы моделируем поведение отдельной птицы (или, по крайней мере, ту часть поведения птицы, которая позволяет ей участвовать в стае). Чтобы поддержать эту поведенческую «структуру управления», мы должны также смоделировать части механизмов восприятия птиц и аспекты физики аэродинамического полета. Если эта



смоделированная модель птицы имеет правильное поведение членов стаи, то всё, что потребуется для создания смоделированной стаи, — это создать несколько экземпляров смоделированной модели птицы и позволить им взаимодействовать [1].

Чуть позже в “Boids Background and Update” [2] Рейнольдс даст три основных критерия поведения каждого отдельно взятого объекта:

- Разделение: держитесь подальше от скопления местных товарищей по стае.
- Выравнивание: держитесь среднего курса местных товарищей по стае.
- Сплоченность: старайтесь двигаться к среднему положению местных товарищей по стае. [2]

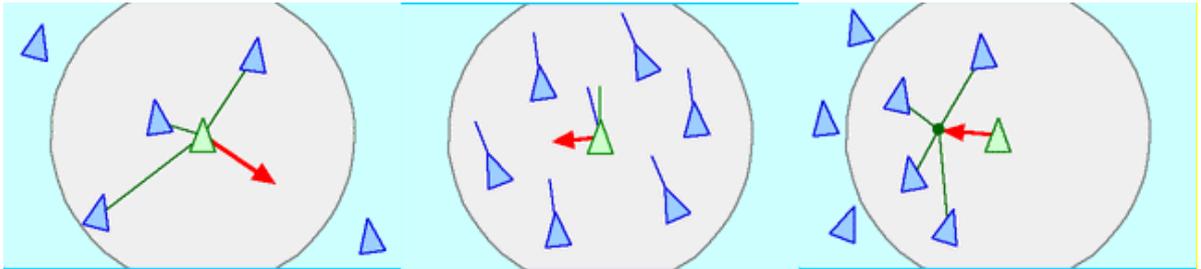


Рис. 1. Иллюстрации Крейга Рейнольдса к основным принципам боидов

### Задача и среда разработки

В качестве задачи для данной публикации, повторим опыт К. Рейнольдса, но с дополнением и улучшениями Джереми Гибсона Бонда, которые включают в себя объект следования, а также более гибкое изменение параметров моделируемой системы [3]. В качестве языка программирования для реализации проекта был выбран C# [4] и графический движок Unity [5] по ряду причин. Во-первых, C# сам выделяет и очищает динамическую память, во-вторых для реализации контейнеров в самом C# в Unity уже присутствует все необходимое. Также Unity обладает удобным для работы представлением трехмерного пространства и векторными операциями.

### Реализация

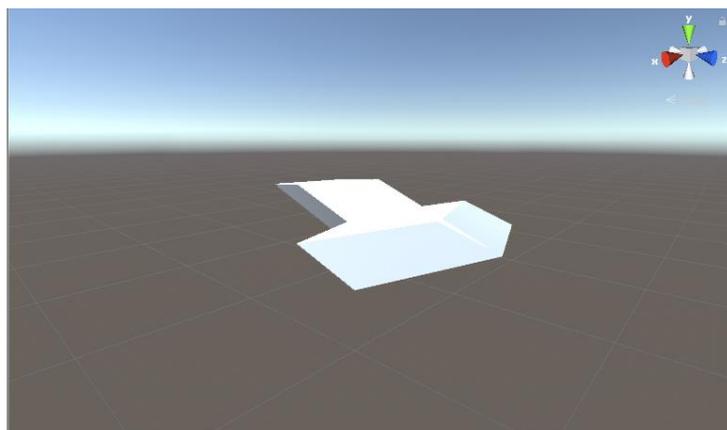


Рис. 2. Трехмерная модель боида в среде Unity

Основным объектом, всей программы является боид. Боид – это объект, в который заложены несколько классов типа MonoBehaviour: Void и Neighborhood. Непосредственно класс Void отвечает за движение за объектом, если расстояние до объекта больше нужного, или от объекта, в ином случае. Neighborhood отвечает за количество допустимых соседей, если соседей более допустимого количества, боид будет стараться избегать их.



У класса `Neighborhood` есть свойство `avgPos`, которое доступно только для чтения, и просматривает все экземпляры класса `Void` в списке `neighbors` и вычисляет координаты центральной точки в группе. [3]

Листинг 1 – `avgPos` в классе `Neighborhood`

```
public Vector3 avgPos{
    get {
        Vector3 avg = Vector3.zero;
        if (neighbors.Count == 0) return avg;
        for (int i=0; i < neighbors.Count; i++){
            avg += neighbors[i].pos;
        }
        avg /= neighbors.Count;
        return avg;
    }
}
```

Сами “птицы” создаются классом-спавнером, который логически связан с главной камерой. В нем в каждый вызов функции ядра `Unity FixedUpdate()` создаются объект, который наделяется соответствующими параметрами, которые можно редактировать в среде `Unity`.

Объект следования `Attractor` представляет собой простой круг, движущийся по синусоиде. Если птица достаточно далеко от центра притяжения `Attractor` и стремится к нему, то вызовом функции `Lerp()` выполняется линейная интерполяция `vel` (птицы) в направлении `velAttract` (аттрактор). Поскольку векторы `vel` и `velAttract` имеют одинаковую магнитуду (длину), интерполяция производит равномерное взвешивание. На рисунке 3 показано, что, если птица оказалась слишком близко от точки `Attractor.POS`, тогда выполняется линейная интерполяция `vel` в направлении, противоположном `velAttract` [3].

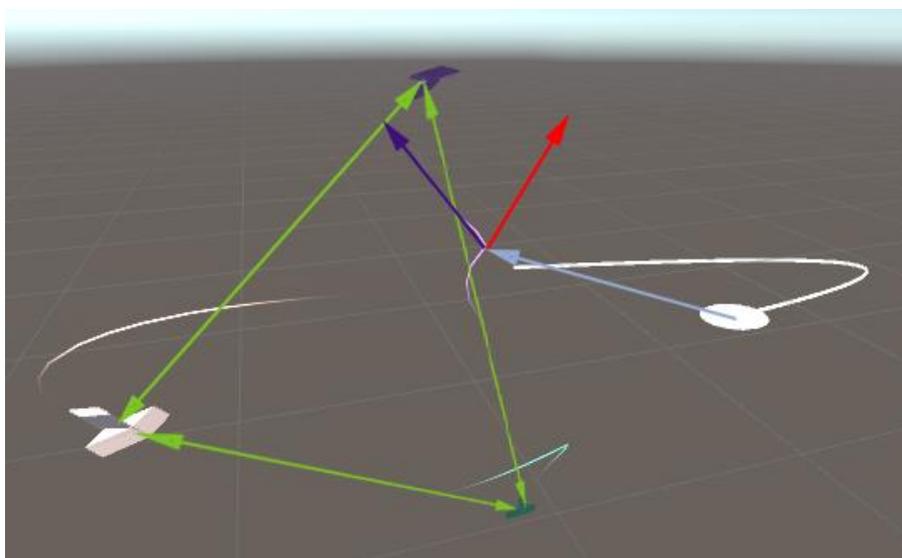


Рис. 3. Поведение трех боидов

Главным изменением Бонда, внесенным в основную программу, является выноска определенных параметров для прямого редактирования. С помощью такой возможности мы можем сами редактировать поведение боидов. Например, заставить их летать ближе к объекту следования или держаться мелкими стаями.

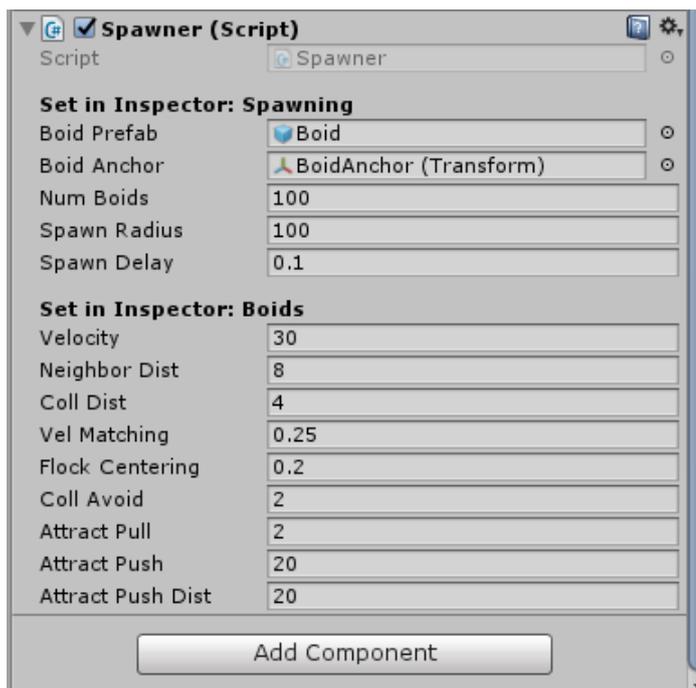


Рисунок 4. Редактирование параметров

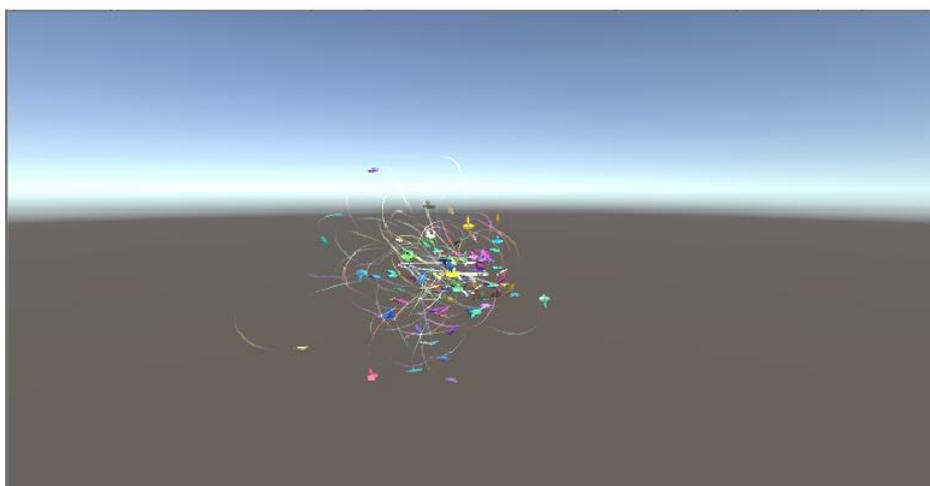


Рисунок 5. Стандартное поведение боидов

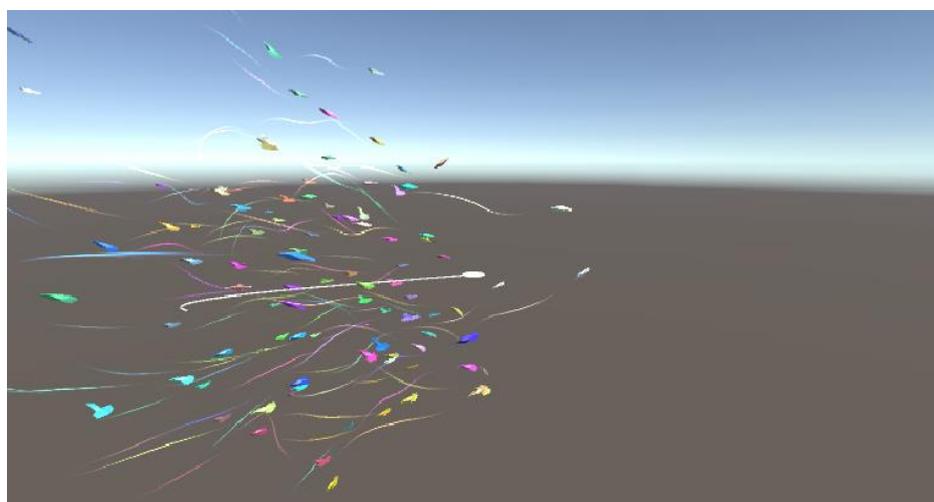


Рисунок 6. Разрозненная стая.



## Заключение

С использованием современных сред разработки Unity и языка программирования C#, была реализована математическая модель, разработанная Крейгом Рейнольдсом в 1987 году. Эта модель включает алгоритм "Боиды" для имитации коллективного поведения объектов. С помощью Unity и C# были реализованы компоненты, которые обрабатывают расположение, скорость и направление боидов на основе их окружающей среды. Такая реализация позволяет создавать симуляции стай птиц, рыб или других коллективных объектов движущихся относительно общего центра.

## СПИСОК ЛИТЕРАТУРЫ

1. Reynolds C. W. Flocks, Herds, and Schools: A Distributed Behavioral Model [Электронный ресурс] // Published in *Computer Graphics*, July 1987. № 21(4). p. 25-34. – URL: <http://www.cs.toronto.edu/~dt/siggraph97-course/cwr87/> (дата обращения 08.09.2023)
2. Reynolds C. W. Boids Background and Update [Электронный ресурс]. – URL: <https://red3d.com/cwr/boids/> (дата обращения 08.09.2023)
3. Д. Г. Бонд. Unity и C# геймдев от идеи до реализации / Д. Г. Бонд; пер. с англ. А. Кисилев, ред. Ю. Сергиенко – Санкт-Петербург: Питер. Второе издание. 2019. – 928 с.
4. Microsoft: Документация по C#. Обзор классов, структур и записей в C # [Электронный ресурс]. – URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/object-oriented/> (дата обращения 08.09.2023)
5. Unity: программирование в Unity [Электронный ресурс]. – URL: <https://unity.com/ru/solutions/programming> (дата обращения 08.09.2023)

## ИНФОРМАЦИЯ ОБ АВТОРАХ

### **Забурдин Вячеслав Дмитриевич –**

студент кафедры аэрокосмических компьютерных программных систем  
Санкт-Петербургский государственный университет аэрокосмического приборостроения  
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А  
E-mail: zaburdinslv@mail.ru

### **Абдрашитов Роман Рустанович –**

Студент кафедры аэрокосмических компьютерных и программных систем  
Санкт-Петербургский государственный университет аэрокосмического приборостроения  
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А  
E-mail: abdrashitov.roman.r@gmail.com

### **Шахомиров Андрей Викторович –**

Доцент кафедры аэрокосмических компьютерных и программных систем, кандидат технических наук  
Санкт-Петербургский государственный университет аэрокосмического приборостроения  
190000, Санкт-Петербург, ул. Большая Морская, д. 67, лит. А  
E-mail: shakhomirov@gmail.com

## INFORMATION ABOUT THE AUTHORS

### **Zaburdin Vyacheslav Dmitrievich –**

student of the Departments of Aerospace Computer Software Systems  
Saint-Petersburg State University of Aerospace Instrumentation  
67, Bolshaya Morskaya str., Saint-Petersburg, 190000, Russia  
E-mail: zaburdinslv@mail.ru



**Abdrashitov Roman Rustemovich –**

Student of the Departments of Aerospace Computer Software Systems  
Saint-Petersburg State University of Aerospace Instrumentation  
67, Bolshaya Morskaia str., Saint-Petersburg, 190000, Russia  
E-mail: abdrashitov.roman.r@gmail.com

**Shakhomirov Andrey Viktorovich –**

Associate Professor of the Department of Aerospace Computer and Program Systems, PhD  
Saint-Petersburg State University of Aerospace Instrumentation  
67, Bolshaya Morskaia str., Saint-Petersburg, 190000, Russia  
E-mail: shakhomirov@gmail.com