

UNITY. ДВИЖЕНИЕ ОБЪЕКТА ПО ЗАДАННОЙ В ПОЛЯРНОЙ СИСТЕМЕ КООРДИНАТ ТРАЕКТОРИИ

И. С. Смоляк, Е. В. Крот

lenakrot0408@gmail.com;

smolakilona0@gmail.com;

Научный руководитель — О. М. Кветко, старший преподаватель

Unity – это кроссплатформенный игровой движок, который позволяет создавать игры и приложения для различных платформ. Один из преимуществ Unity заключается в широком выборе инструментов и функций, которые помогают разработчикам реализовывать сложные задачи, такие как движение объектов в разных системах координат.

В данной статье будут рассмотрены способы движения объекта по заданным в полярной системе координат траекториям. Полярная система координат - это система координат, в которой положение точки определяется с помощью угла и расстояния от начала координат.

Ключевые слова: игровой движок Unity; полярная система координат; движение объектов по заданным траекториям.

Движение объекта будет реализовано по двум траекториям: спираль Архимеда и кривые Безье. Для реализации движения объекта по траекториям в полярной системе координат будет использован *UNITY* и язык программирования *C#*. В качестве объекта был выбран самолет. Для представления полярной системы координат будет создана сетка, на которой располагаются цели, к которым двигается объект.

СПИРАЛЬ АРХИМЕДА

Спираль Архимеда - плоская кривая, траектория точки *M*, которая равномерно движется вдоль луча *OV* с началом в *O*, в то время как сам луч *OV* равномерно вращается вокруг *O* (Рис. 1). Спираль Архимеда задается в полярных координатах уравнением:

$$r = a + b * \varphi, \text{ где } \varphi \in \left[\frac{a}{b}, \infty \right)$$

Все вычисления требуется проводить в декартовых координатах, поэтому полярные координаты переводятся в декартовы с помощью данных формул:

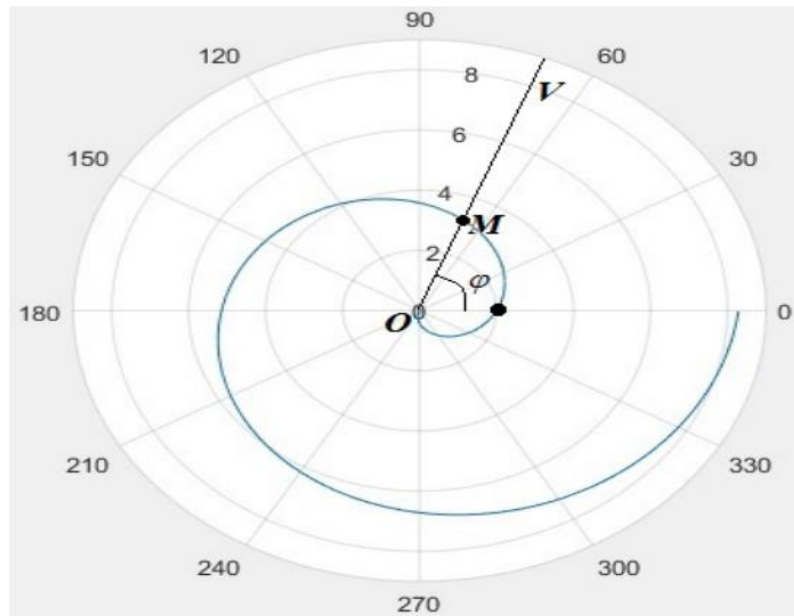
$$x = r * \cos(a),$$

$$y = r * \sin(a),$$

где *x* и *y* - координаты точки, *r* - радиус, *a* – угол [1].

Чтобы определить радиус и угол спирали используются следующие формулы:

$$\text{angle} = a * t, \text{ radius} = b * t [2].$$



Спираль Архимеда

Для создания движения объекта по спирали Архимеда в Unity необходимо его определить и присвоить новый скрипт (*Листинг 1*). Требуется задать 2 константы, отвечающие за контроль угла поворота спирали. Переменная t отвечает за время, которое прошло с начала движения объекта [3].

Сперва задаём формулы в полярных координатах, после переходим к декартовым. Добавляем вращение самолёта.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class spiral : MonoBehaviour
{
    private const float RotationSpeed = 0.33;
    private const float RunAwaySpeed = 0.33f;
    void Update()
    {
        float t = Time.time;
        float radius = RunAwaySpeed * t;
        float angle = RotationSpeed * t;
        float x = Mathf.Cos (angle) * radius;
        float y = Mathf.Sin (angle) * radius;
        this.transform.position = new Vector3 (x,0,y);
        this.transform.rotation = Quaternion.Euler (0, -angle / Mathf.PI *
180, 0);
    }
}
```

}

КРИВЫЕ БЕЗЬЕ

Кривые Безье являются одними из наиболее популярных кривых в компьютерной графике. Они используются для создания плавных и красивых кривых на графическом экране. Кривые Безье определяются набором управляющих точек, которые задают форму кривой. Формула для кривых Безье может быть выражена следующим образом:

$$B(t) = (1 - t)^3 * P0 + 3 * t * (1 - t)^2 * P1 + 3 * t^2 * (1 - t) * P2 + t^3 * P3,$$
 где $P0$ - $P3$ – управляющие точки кривой, t – параметр времени, изменяющийся от 0 до 1. Для создания движения объекта по кривым Безье в Unity необходимо создать два скрипта, в одном из которых описан метод, принимающий четыре точки и параметр t . Метод возвращает координату точки на кривой Безье (*Листинг 2*).

Второй скрипт прикрепляется к объекту, в нём задаются четыре точки, параметр t и применяется метод, реализованный в скрипте *Bezier*.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public static class Bezier
{
    public static Vector3 GetPoint(Vector3 p0, Vector3 p1, Vector3 p2, Vector3 p3, float t)
    {
        Vector3 p01 = Vector3.Lerp(p0, p1, t);
        Vector3 p12 = Vector3.Lerp(p1, p2, t);
        Vector3 p23 = Vector3.Lerp(p2, p3, t);
        Vector3 p012 = Vector3.Lerp(p01, p12, t);
        Vector3 p123 = Vector3.Lerp(p12, p23, t);
        Vector3 p0123 = Vector3.Lerp(p012, p123, t);
        return p0123;
    }
}
```

РЕЗУЛЬТАТЫ

Было создано две сцены, в которых объект двигался по заданным траекториям в полярной системе координат. Изначально было установлено несколько целей на сетке с заданными координатами. При нажатии на кнопку старт самолет начинал движение по заданной траектории к цели.

В этой статье было рассмотрено, как в *Unity* можно реализовать движение объекта по заданным траекториям в полярной системе координат, таким как спираль Архимеда и кривые Безье. Для реализации движения

были использованы скрипты, которые позволяют определить траекторию движения объекта с помощью параметров и управляющих точек. Эти скрипты могут быть дополнены и изменены в соответствии с требованиями конкретного проекта, что делает *Unity* еще более мощной и гибкой платформой для разработки игр и приложений.

Реализация движения объекта по траектории в полярной системе координат была успешно выполнена. В дальнейшем эту технологию можно улучшить за счет добавления дополнительных элементов, таких как звуковые эффекты и анимации.

Библиографические ссылки

1. Unity и C# / Д. Бонд. СПб: Питер, 2019. 930 с.