

УДК 004.946

ЗАСОБИ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ У КОМП'ЮТЕРНІЙ ГРІ

Кузьменко П.М.

e-mail: pylyp.kuzmenko@nure.ua

Науковий керівник – д.т.н., проф., Гороховатський В.О.

Харківський національний університет радіоелектроніки, каф. ІНФ
м. Харків, Україна

The work explores the utilization of virtual reality tools in a computer game made with Unity engine, including but not limited to built-in XR Interaction Toolkit, Oculus XR Plugin, OpenXR Plugin, AI navigation, Input System, Shader Graph and Universal Render Pipeline packages; and C# programming language where player can handle different weapons and test them in various scenarios such as shooting range. Game features projectile ballistics, realistic optical scopes, FPV drone control and, possibly in the future, AI based on Goal-Oriented Action Planning (GOAP) algorithm.

Робота досліджує використання інструментів віртуальної реальності в комп'ютерній грі, створеній за допомогою рушія Unity, включаючи, але не обмежуючись, вбудованими пакетами XR Interaction Toolkit, Oculus XR Plugin, OpenXR Plugin, AI navigation, Input System, Shader Graph та Universal Render Pipeline; та застосуванням мови програмування C#. Гравець може працювати з різними моделями зброї та випробовувати її в різних сценаріях, таких як стрільбище. У грі є моделювання балістики снарядів, реалістичні оптичні приціли, керування FPV безпілотником та, можливо в майбутньому, засоби штучного інтелекту на основі алгоритму Goal-Oriented Action Planning (GOAP).

Для реалізації взаємодії гравця із віртуальним середовищем застосовується пакет Input System. Цей пакет є новою, більш гнучкою системою, яка дозволяє використовувати будь-який пристрій введення для керування вмістом гри. Він призначений для заміни класичної системи введення Unity.

Створено контролер FPV дрона. Визначимо вектори $a = (a_x, a_y)$ та $b = (b_x, b_y)$, що позначають введення гравця на стіках для великого пальця, $i = (i_x, i_y, i_z)$ – позиція мотора відносно дрона, m – множник сили, а $f(a)$ та $g(b)$ – криві, що виражають ступінь відклику на дію користувача. Визначимо силу F , що створена моторами дрона, вона розподіляється контролером по моторам дрона за наступним принципом:

$$F = f(a_y \cdot 0.5 + 0.5)m(1 - x)(1 - y)$$
$$x = \{0, \text{якщо } i_x b_x \leq 0 \ g(|b_x|)$$
$$y = \{0, \text{якщо } i_z b_y \leq 0 \ g(|b_y|)$$

Для моделювання дій штучного інтелекту в даному ігровому проєкті планується застосувати алгоритм Goal-Oriented Action Planning (GOAP).

Цей алгоритм є похідним від алгоритму STRIPS (STanford Research Institute Problem Solver), який складається з цілей та дій, де цілі описують певний бажаний стан світу, якого ми хочемо досягти, а дії визначаються з точки зору передумов та ефекту [1]. Використання такого методу моделювання поведінки штучного інтелекту має низку переваг, перша перевага – це здатність розділити цілі та дії, дозволити різним типам персонажів задовольняти цілі різними способами. Друга перевага системи планування полягає в полегшенні шарування простих моделей поведінки для створення складної спостережуваної поведінки. Третя перевага – наділення персонажів здатністю динамічного вирішення проблем [2].

Ігровий проєкт застосовує інструменти пакету Visual Effects Graph. Пакет Visual Effects Graph є візуальним рішенням на основі вузлів, що дає змогу розробляти візуальні ефекти, від простої поведінки частинок до складних симуляцій, що виконуються на графічному процесорі. Прикладом застосування цього інструменту в ігровому проєкті є ефект виходу порохів газів зі ствола зброї. Так застосовуються наступні візуальні вузли: Spawn – цей вузол визначає правила появи нових частинок; Initialize Particle – ініціалізує нові частинки та задає такі параметри як позиція, орієнтація у просторі, швидкість та час життя частинок; Update Particle – визначає які ефекти діють на частинки кожного кроку симуляції; Output Particle Unlit – визначає візуальні аспекти частинок, такі як орієнтація у просторі, залежність розміру та кольору частинок від часу їхнього життя.

Проєкт застосовує інструменти пакету Shader Graph. Shader Graph – це інструмент, який дає змогу візуально створювати шейдери. Замість написання коду, користувач пакету з'єднує вузли в структурі графа, таким чином створюючи різні моделі відображення об'єктів. Цей інструмент надає миттєве відображення змін внесених у граф. В даному ігровому проєкті шейдер оптичного прицілу реалізовано через інструменти Shader Graph, так базовий колір фрагментного шейдера вираховується шляхом покомпонентного перемноження сили віньєтки затінення та зображення в прицілі [3–5]. Віньєтка затінення вираховується шляхом ділення відстані від оброблюваного пікселя до віртуальної камери на параметри оптимальної дистанції погляду та радіуса віньєтки, переведення координат у полярні з радіальним масштабом у обчислене значення, піднесенням до певного степеня, що визначає м'якість віньєтки, та відніманням одиниці від обчисленого значення. Зображення в прицілі вираховується шляхом застосування функції лінійної інтерполяції Lerp за зображенням з віртуальної камери прицілу та зображенням сітки прицілу, по альфа каналу останнього зображення. Для семплінгу текстур також застосовуються власні перетворення UV координат зі зміщенням, що вираховується шляхом віднімання глобальних координат оброблюваного пікселя від глобальних координат камери, переведення напрямку з глобального до дотичного простору та множення отриманого значення на параметр масштабу.

В підсумку ми маємо розроблений ігровий проєкт, що успішно застосовує інноваційні засоби рушія Unity та пакетів XR Interaction Toolkit, Input System, Shader Graph та Universal Render Pipeline, що дозволяє покращити користувацький досвід при взаємодії з інтерактивними об'єктами у віртуальній реальності. Окрім цього проєкт успішно застосовує інструменти легкого у плані використання ресурсів конвеєра візуалізації Universal Render Pipeline.

Список використаних джерел:

1. Nilsson, N. J. (1998). STRIPS Planning Systems. *Artificial Intelligence: A New Synthesis*, 373-400. Morgan Kaufmann Publishers, Inc.
2. Orkin, J. (2006). Three States and a Plan: The A.I. of F.E.A.R. *Game Developers Conference 2006*.
3. Daradkeh, Y.I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., and AIDhaifallah, M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, 11, 126938-126949.
4. Gadetska, S.V., Gorokhovatskyi, V. O., Stiahlyk, N. I., Vlasenko, N.V. Statistical data analysis tools in image classification methods based on the description as a set of binary descriptors of key points. *Radio Electronics, Computer Science, Control*, 2021, №4, pp. 58-68.
5. Gorokhovatskyi, V., Tvoroshenko, I., Yakovleva, O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, 33 (1), 113-125.