

УДК 004.9

*Мельников О.Ю., к.т.н., доцент
Пархоменко Ю.Д., студент*

*Донбаська державна машинобудівна академія, м. Краматорськ,
Україна*

РОЗРОБКА МОДУЛЬНОЇ СИСТЕМИ КЕРУВАННЯ ПОВЕДІНКОЮ КАМЕРИ В СЕРЕДОВИЩІ UNITY

У сучасних ігрових проєктах, що розробляються в середовищі Unity, система камери є одним із ключових компонентів, що безпосередньо впливає на зручність взаємодії користувача з грою [1]. У більшості випадків реалізація поведінки камери виконується як монолітний скрипт, що ускладнює його розширення, повторне використання та підтримку при зростанні складності проєкту [3]. Це особливо актуально для ігор, у яких камера повинна підтримувати декілька режимів роботи, таких як слідування за об'єктом та вільне переміщення.

Було поставлено й вирішено задачу розробки модульної системи поведінки камери, яка дозволяє гнучко комбінувати різні режими роботи, джерела вхідних даних та алгоритми руху. Запропонований підхід базується на принципах компонентної архітектури та розділення відповідальностей [3-4]. Система включає окремі модулі для визначення режиму роботи камери, отримання вхідних даних користувача та цільових позицій, а також обчислення кінцевого положення камери у просторі.

Однак при розробці такої системи виникає наступне завдання: забезпечити універсальний механізм взаємодії між різними режимами камери, зберігаючи при цьому простоту використання та розширення. Формально це можна описати як задачу побудови абстрактного інтерфейсу ICameraMode, що визначає базові операції керування станом камери, а також набору допоміжних інтерфейсів для джерел даних (позиція цілі, користувацький ввід) та алгоритмів переміщення [4].

Запропонована архітектура передбачає розділення системи на три основні компоненти: режими камери (Follow, Free Move), постачальники даних (Input Provider, Target Provider) та модулі руху (Mover). Кожен режим визначає логіку високого рівня, використовуючи відповідні джерела даних і алгоритми обчислення позиції. Це дозволяє легко додавати нові режими або змінювати поведінку існуючих без модифікації інших частин системи [3, 5].

Пропонується наступний алгоритм функціонування системи. На кожному кадрі активний режим камери отримує необхідні дані

(позицію цілі або напрямок руху), передає їх у модуль обчислення руху, після чого застосовує результат до трансформа камери. Перемикання між режимами здійснюється через централізований контролер, що забезпечує коректну ініціалізацію та завершення роботи кожного режиму [1-2].

Використаємо наступні допущення:

- камера розглядається як об'єкт із позицією та орієнтацією у тривимірному просторі;
- вхідні дані користувача нормалізовані та не залежать від конкретного пристрою введення;
- цільова позиція для режиму слідування визначається або через Transform, або як фіксована точка у просторі.

У подальшому необхідно реалізувати запропоновану модель у середовищі програмування та оцінити її ефективність з точки зору масштабованості, зручності використання та продуктивності [5-6].

Список використаних джерел

1. Unity Technologies. Unity User Manual. URL: <https://docs.unity3d.com/Manual/index.html> (дата звернення: 24.03.2026).
2. Unity Technologies. Cinemachine Documentation. URL: <https://docs.unity3d.com/Packages/com.unity.cinemachine@latest> (дата звернення: 24.03.2026).
3. Robert C. Martin Clean Architecture: A Craftsman's Guide to Software Structure and Design. – Boston: Prentice Hall, 2017. – 432 с.
4. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides Design Patterns: Elements of Reusable Object-Oriented Software. – Boston: Addison-Wesley, 1994. – 395 с.
5. Jason Gregory Game Engine Architecture. – Boca Raton: CRC Press, 2018. – 1136 с.
6. Ian Millington Game Physics Engine Development. – Boca Raton: CRC Press, 2010. – 512 с.