

УДК 004.4

*Зулінський М.В., здобувач,**Марчук Д.К., ст. викладач**Державний університет «Житомирська політехніка»*

ПІДХІД ДО ПОБУДОВИ МОДУЛЬНОЇ АРХІТЕКТУРИ ІГРОВОГО ПРОЦЕСУ НА РУШІІ UNITY

Сучасні підходи до розробки ігор вимагають високий рівень структурованості коду. Через зростання складності в побудові ігор, за рахунок збільшення можливих механік призводить до неможливості покращувати код і змінювати його певні аспекти без шкоди. Тому, доцільно застосовувати підхід з розділенням більш складних задач на автономні частини – модулі [1]. Модуль виконує лише задані для нього функції та підтримується як інтерфейс, що забезпечить легкість в адаптації та підтримці коду, а також у тестуванні. В Unity цей підхід дуже чудово вписується через принцип роботи рушія та його орієнтацію на об'єктно орієнтовану систему взаємодії та компоненти, що робить можливість інтеграції даного підходу дуже легким у використанні.

Метою дослідження являється доведення переваг модульної архітектури у складних проектах, та важливість її застосування на рушії Unity.

Unity через свою орієнтацію на об'єктно-компонентну систему надає нам вже готові засоби для інтеграції модульного підходу, що спирається на компонентні моделі. У такій моделі `GameObject` виступає контейнером для набору компонентів, які визначають окремі аспекти поведінки [2]. На практиці це відкриває шлях до використання різних підходів і інструментів, що допомагають будувати масштабовану та гнучку архітектуру. Серед найпоширеніших варто виділити такі:

1. *ScriptableObject* використовується для зберігання даних проекту та окремої логіки, яку можна повторно застосовувати на різних сценах і об'єктах без дублювання чи перевизначення коду [3].

2. *Event-Drive Design* забезпечує взаємодію модулів через події, що мінімізує залежності між ними. Наприклад, система здоров'я може надіслати повідомлення про зміну стану, не знаючи нічого про інші модулі.

3. *Entity Component System (ECS)* використовується у великих проектах для розділення поведінки та даних для подальшого опрацювання великої кількості об'єктів, одночасно використовуючи вже створені модулі.

4. *Dependency Injection (ID)* відповідає за передачу залежностей між модулями без жорстокої прив'язки до конкретних компонентів та

класів, що в свою чергу підвищує можливість тестування та спрощує внесення змін до всього модуля.

На практиці модульний підхід у розробці ігрових процесів на Unity дозволяє створювати архітектуру, в якій окремі елементи гри можна легко змінювати та розширювати без втручання в основний або залежний код. Це спрощує адаптацію проекту під різні платформи, зменшує дублювання логіки, покращує повторне використання компонентів та знижує ризик появи помилок у пов'язаних модулях.

До прикладу в іграх жанру RPG модулі можуть бути розділені та відповідати за керування інвентарем, бойовими системами, штучним інтелектом а також взаємодією з UI елементами. Завдяки такому поділу зміни в одному модулі не порушують роботу інших і не впливають на їхній функціонал. Таким чином алгоритми атаки для ворогів не мають впливу на UI та інвентар, а також не впливають на бойову систему.

Використання модульної архітектури являється ключовим підходом для побудови сучасних ігор. Вона впливає на гнучкість, масштабованість та подальшу підтримку проекту, що легко адаптується при потребі. Використання вже готових способів застосування компонентних моделей в Unity впливає на час розробки коду та його якість. Таким чином, модульна архітектура не лише впливає на покращення організації коду, а також збільшує ефективність команди при розділенні задач між ними, що є критично важливим у сучасній ігровій індустрії.

Список використаних джерел:

1. Modular Software: Definition, Benefits, and Architecture Principles. *vFunction Blog*. 2023. URL: <https://vfunction.com/blog/modular-software/> (дата звернення: 19.11.2025).
2. Nordon S. Unity Architecture: GameObject-Component Pattern. *Medium*. 2022. URL: <https://medium.com/@simon.nordon/unity-architecture-gameobject-component-pattern-34a76a9eacfb> (дата звернення: 20.11.2025).
3. Create Modular Game Architecture with ScriptableObjects in Unity 6. *Unity Resources*. 2024. URL: <https://unity.com/resources/create-modular-game-architecture-scriptableobjects-unity-6> (дата звернення: 20.11.2025).