

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ МОДУЛЬНОЇ АРХІТЕКТУРИ ІГРОВОГО ЗАСТОСУНКУ В СЕРЕДОВИЩІ UNITY ІЗ ЗАСТОСУВАННЯМ ПАТЕРНІВ ПРОЄКТУВАННЯ

Вінницький національний технічний університет

Анотація

Розглянуто особливості проєктування та програмної реалізації модульної архітектури тривимірного ігрового застосунку в жанрі стратегії / Tower Defense. Програмну реалізацію виконано в ігровому рушії Unity з використанням об'єктно-орієнтованої мови програмування C# та підсистеми просторової навігації NavMesh. Проаналізовано актуальність впровадження класичних патернів проєктування (Gang of Four) для уникнення сильної зв'язності коду, підвищення гнучкості архітектури та оптимізації ігрового циклу в реальному часі.

Ключові слова: модульна архітектура, патерни проєктування, Unity, C#, штучний інтелект, NavMesh, Tower Defense, програмна інженерія.

Abstract

The features of design and software implementation of a modular architecture for a 3D game application in the strategy / Tower Defense genre are considered. The software implementation is performed in the Unity game engine using the object-oriented C# programming language and the NavMesh spatial navigation subsystem. The relevance of implementing classic design patterns (Gang of Four) to avoid tight coupling of code, increase architectural flexibility, and optimize the game loop in real time is analyzed.

Keywords: modular architecture, design patterns, Unity, C#, artificial intelligence, NavMesh, Tower Defense, software engineering.

Вступ

У сучасних умовах стрімкого розвитку індустрії інтерактивних розваг, комп'ютерних ігор та симуляційних систем ключову роль відіграє не лише візуальна складова, а й чистота, масштабованість та швидкодія архітектури програмного забезпечення [1]. Базові підходи до розробки ігрових програм часто спираються на монолітні класи, що призводить до антипатерну “Spaghetti code” та критично високої зв'язності компонентів. Це суттєво ускладнює розширення системи при впровадженні складних алгоритмів, таких як процедурна генерація простору або автономна навігація штучного інтелекту. Актуальність розробки програмного модуля ігрового застосунку з модульною архітектурою полягає в необхідності побудови слабко зв'язаної структури, яка дозволяє гнучко налаштовувати ігрову логіку, ізольовано тестувати механіки стратегічного будівництва і розгортання пасток, а також раціонально розподіляти навантаження на обчислювальні ресурси ЕОМ[2].

Метою дослідження є забезпечення високої гнучкості програмних компонентів, зниження складності супроводу коду та демонстрація практичної ефективності інтеграції об'єктно-орієнтованих шаблонів проєктування в екосистему рушія Unity.

Особливості розробки модульної архітектури ігрового застосунку

Для реалізації програмного комплексу обрано ігровий рушій Unity та мову програмування C#, що забезпечує повний спектр інструментів для об'єктно-орієнтованого моделювання і побудови Data-Driven архітектури. Використання C# дозволяє впровадити класичні шаблони проєктування GoF. Зокрема, патерн *Singleton* застосовано для централізованих менеджерів (*GameManager*, *PlacementManager*), що гарантує єдину точку доступу до економіки та станів гри. Шаблон *Factory Method* забезпечує гнучке створення хвиль ворожих агентів без жорсткої прив'язки до їхніх конкретних класів. Подієва модель на базі патерну *Observer* виступає основою для реактивного оновлення інтерфейсу (HUD) при зміні балансу золота чи здоров'я, повністю усуваючи необхідність постійного опитування даних у ресурсомістких циклах оновлення.

Завдяки інтеграції підсистеми Unity AI Navigation (NavMesh), автономні ігрові персонажі самостійно прокладають оптимальні маршрути у процедурно згенерованому лабіринті, побудованому за алгоритмом пошуку в глибину (DFS). Застосування нової системи вводу Unity Input System дозволило створити адаптивний контролер ізометричної камери з можливостями плавного масштабування та обертання, що суттєво покращує користувацький досвід (UX) [3]. Всі базові конфігурації характеристик споруд винесені у незалежні файли даних ScriptableObject, відокремлюючи ігровий баланс від безпосереднього програмного коду.

Розроблена архітектурна модель підтримує динамічне асинхронне перерахування навігаційних графів у реальному часі. Коли гравець розміщує нову будівлю на сцені, система проводить фізичну валідацію простору через метод Physics.OverlapBox і миттєво оновлює локальну ділянку NavMesh-сітки. Це змушує ШІ-агентів оперативно змінювати траєкторію руху в обхід нової перешкоди. Важливою перевагою такого підходу є оптимізація розподілу обов'язків між класами, що розвантажує обчислювальні потоки, запобігає витокам оперативної пам'яті та забезпечує стабільну частоту кадрів (60+ FPS).

Основними функціональними можливостями розробленого модуля є:

- процедурна генерація 3D-лабіринтів на основі математичного алгоритму DFS із механізмом Backtracking;
- стратегічне розміщення захисних споруд із жорсткою прив'язкою до сітки (Grid Snapping) та перевіркою фізичних колізій;
- автономна маршрутизація та адаптивний пошук шляху ШІ-агентами за допомогою динамічного запікання NavMesh;
- гнучке керування ігровим циклом та економічною моделлю (нарахування та списання ресурсів) через систему подій;
- ізометричне керування та огляд поля бою за допомогою оптимізованого контролера камери.

Проведений аналіз типових ігрових шаблонів та аналогів (зокрема, Unity TD Template) показав, що їхня архітектура часто страждає від надмірної зв'язності компонентів. Логіка інтерфейсу в них може безпосередньо керувати станом ядра програми, що унеможливує безпечне масштабування та ізольоване тестування. Саме тому розробка власної модульної системи із застосуванням чітких абстракцій та патернів проектування дозволяє повністю розділити рівень даних, рівень бізнес-логіки та рівень представлення, утворюючи надійне, прозоре та вузькоспеціалізоване інженерне рішення.

До основних переваг розробленого застосунку можна віднести:

- повний контроль над життєвим циклом об'єктів та структурою взаємодії ігрових підсистем;
- низьку зв'язність класів, що значно полегшує процес розширення та додавання нових ігрових сутностей;
- високу продуктивність графічного конвеєра та фізичного рушія за рахунок оптимізації викликів;
- зручність конфігурування геймплейних параметрів через вікно Інспектора Unity та об'єкти ScriptableObject;
- мінімізацію виникнення критичних архітектурних помилок під час масових ігрових подій (хвиль ворогів).

Висновок

У результаті виконано аналіз актуальності проектування модульних архітектур ігрових систем та визначено основні переваги впровадження шаблонів проектування GoF для оптимізації ігрового циклу. Розроблений програмний модуль у середовищі Unity за допомогою мови C# забезпечує ефективне функціонування гри з можливістю процедурної генерації простору, валідації фізичних зон та автономної навігації ШІ. Отримані результати підтверджують доцільність обраного підходу для створення гнучких інтерактивних застосунків, мінімізації зв'язності коду та розширення архітектурних можливостей сучасних ігрових рушіїв.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Game Developer. Overcoming Spaghetti Code in Game Development [Електронний ресурс] – URL: <https://www.gamedeveloper.com/programming/overcoming-spaghetti-code> (дата звернення: 10.02.2026).
2. GeeksforGeeks. Software Design Patterns [Електронний ресурс] – URL: <https://www.geeksforgeeks.org/software-design-patterns/> (дата звернення: 12.02.2026).

3. Unity Documentation. Unity User Manual (2022.3 LTS) [Електронний ресурс] – URL: <https://docs.unity3d.com/Manual/index.html> (дата звернення: 16.02.2026).

Руденко Артем Олександрович – студент групи 5ПІ-22б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, email: artemize@gmail.com

Науковий керівник: **Людмила Броніславівна Ліщинська** – доктор технічних наук, професор кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця, email: llb@vntu.edu.ua

Rudenko Artem – student of group 5PI-22b, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia.

Lishchynska Lyudmyla Bronislavivna – Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: llb@vntu.edu.ua