

ВИКОРИСТАННЯ BLUEPRINT У РОЗРОБЦІ ІНТЕРАКТИВНИХ СИСТЕМ У UNREAL ENGINE 4

Макогон Ю.М., Іщенко Д.К.

e-mail: yurii.makohon1@nure.ua, daria.ishchenko@nure.ua

Науковий керівник – к.т.н., проф. Зубков О.В.

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

The paper examines the use of the Blueprint visual programming system in the Unreal Engine 4 environment for creating interactive game systems. The main focus is on implementing interaction logic between scene objects, handling events, and creating dynamic gameplay elements without using traditional programming code. Within the scope of the work, the capabilities of Blueprint for implementing various types of mechanics are explored, including character interaction with objects, triggering events through trigger zones, creating an object spawning system, and organizing logical scenarios within a game level. The result of the work is the development of a prototype interactive system that demonstrates the effectiveness of using Blueprint in the development of game logic and interactive environments.

Сучасні ігрові рушії надають широкий набір інструментів для створення інтерактивних цифрових середовищ. Одним із ключових інструментів Unreal Engine 4 [1] є система візуального програмування Blueprint, яка дозволяє реалізовувати складну логіку поведінки об'єктів без використання традиційних мов програмування (рисунок 1). Дослідження інновацій цих інструментів є актуальним для розробників ігор.

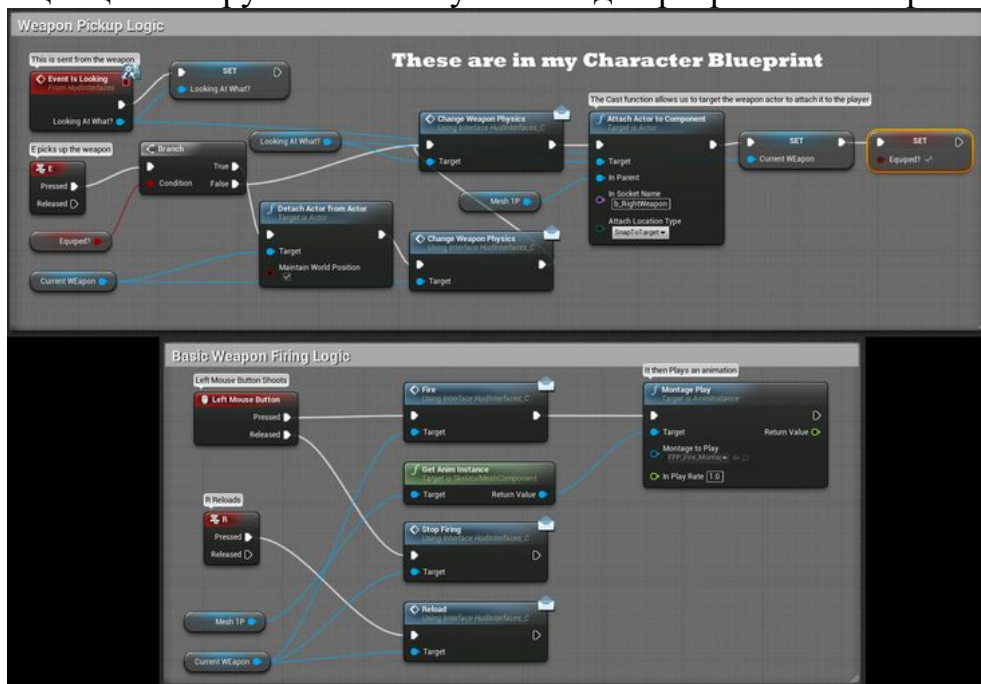


Рисунок 1 – Інтерфейс редактора Blueprint у Unreal Engine 4

Blueprint базується на графічному представленні програмної логіки у вигляді вузлів (nodes) та зв'язків між ними [2,3]. Такий підхід значно спрощує процес створення ігрових механік, дозволяє швидко тестувати різні сценарії взаємодії та зменшує складність розробки для розробників.

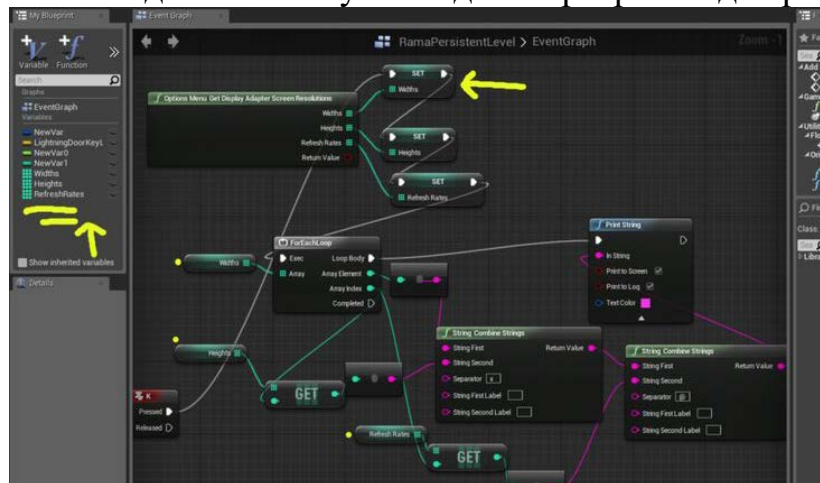


Рисунок 2 – Приклад реалізації логіки у Blueprint Event Graph

У межах проєкту система Blueprint використовується для реалізації взаємодії між персонажем і елементами сцени, обробки тригерних подій та керування поведінкою об'єктів у середовищі. Зокрема, передбачається використання тригерних зон для запуску подій, активації механізмів та зміни стану об'єктів у відповідь на дії користувача (рисунок 3).

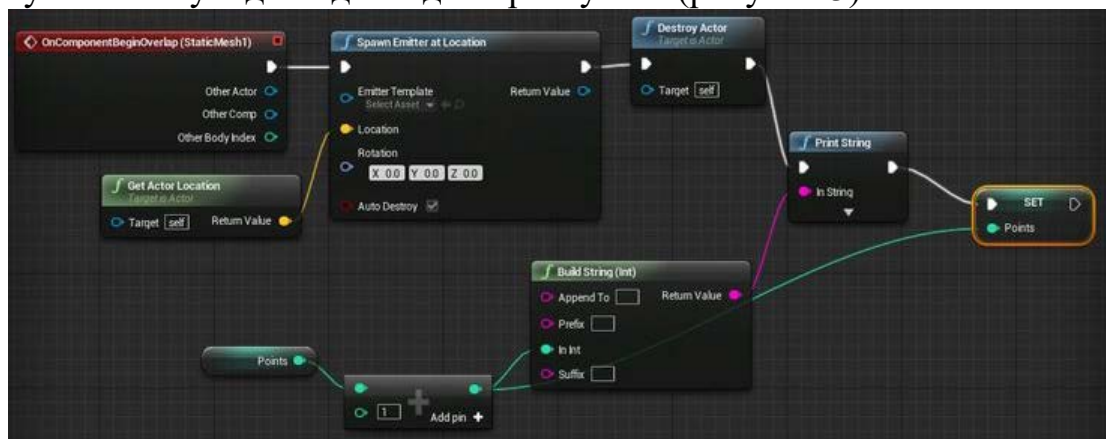


Рисунок 3 – Використання Trigger Volume для запуску подій

Окрему увагу приділено реалізації системи створення об'єктів (spawn system), що дозволяє генерувати нові елементи сцени або відновлювати персонажа у визначених точках. Такий підхід забезпечує гнучке керування ігровим процесом та дозволяє організувати структуровану логіку проходження рівня (рисунок 4). Використання Blueprint також дозволяє реалізувати різноманітні інтерактивні елементи, зокрема рухомі платформи, механізми відкриття дверей, активацію об'єктів та інші сценарії взаємодії. Комбінування таких механік формує комплексну систему геймплею, що реагує на дії гравця та створює динамічне ігрове середовище.

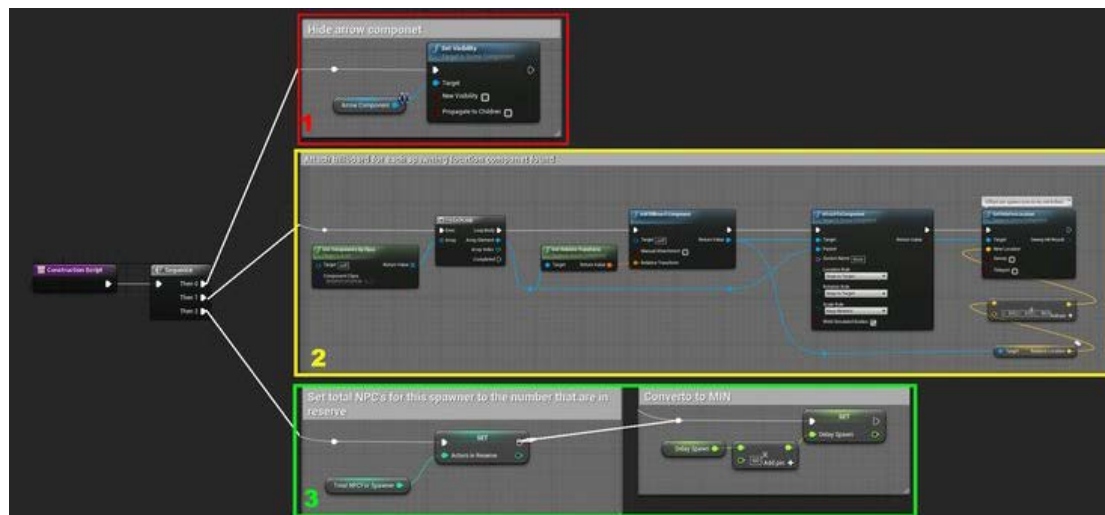


Рисунок 4 – Реалізація створення об'єктів за допомогою Blueprint (Spawn Actor)

Таблиця 1 – Приклади використання Blueprint у проєкті

Елемент системи	Принцип реалізації	Очікуваний результат
Trigger-зони	Використання Blueprint подій	Запуск сценаріїв при вході в зону
Spawn system	Створення об'єктів через Blueprint	Динамічна поява об'єктів у сцені
Інтерактивні об'єкти	Логіка взаємодії через вузли Blueprint	Реакція об'єктів на дії гравця
Рухомі платформи	Анімація та логіка переміщення	Динамічні елементи середовища
Події сценарії	Комбінування вузлів Blueprint	Створення інтерактивного геймплею

Висновки. У роботі було розглянуто можливості використання системи Blueprint у середовищі Unreal Engine 4 для створення інтерактивних ігрових систем. Використання візуального програмування дозволяє ефективно реалізовувати логіку взаємодії між об'єктами сцени, організовувати події сценарії та створювати різноманітні геймплейні механіки. Застосування Blueprint значно спрощує процес розробки, забезпечує гнучкість у створенні інтерактивних систем та дозволяє швидко тестувати різні варіанти реалізації ігрової логіки. Отримані результати демонструють ефективність використання цього інструменту для створення сучасних інтерактивних цифрових середовищ.

Список використаних джерел:

1. Epic Games. Unreal Engine 4 Documentation. – Режим доступу: <https://docs.unrealengine.com> (дата звернення: 10.03.2026)
2. Gregory J. Game Engine Architecture. – CRC Press, 2018.
3. Unreal Engine Blueprint Visual Scripting Guide. – Epic Games, 2020.