

## **ДОСЛІДЖЕННЯ ПРОЦЕСІВ ВИКОРИСТАННЯ ПАТЕРНІВ ПРОЄКТУВАННЯ ПРИ РОЗРОБЦІ ІГОР**

Патерни об'єктно-орієнтованих систем – нелегка справа, а якщо їх потрібно використати повторно, то все стає ще важче. Дизайн таких систем, з одного боку, повинен відповідати задачі, яку система вирішує, а з іншого – бути загальним і абстрактним, щоб врахувати усі вимоги, що можуть виникнути у майбутньому.

Насамперед, досвідченому розробнику зрозуміло, що не потрібно вирішувати кожну нову задачу з нуля. Замість цього, він намагається повторно використати ті рішення, що виявились успішними попередньо. Саме завдяки накопиченому досвіду розробник, що проєктує системи, і стає експертом у своїй області.

Патерни проєктування спрощують повторне використання вдалих проєктних та архітектурних рішень. За рахунок патернів можна покращити якість документації і підтримку існуючих систем за допомогою явного опису взаємодії класів та об'єктів, а також причин, чому система була спроектована саме так, а не інакше.

При розробці системи важливим аспектом є вибір мови та платформи програмування. Це зв'язано з тим, що від мови та стилю залежить ступінь тяжкості реалізації патернів. Саме тому, є сенс виділити розробку ігор як окрему тему для використання патернів.

При розробці гри з віртуальним світом, який містить велику кількість об'єктів, важливим етапом проєктування та розробки є використання патернів. Це дає змогу створити високопродуктивну систему з можливістю розширення.

Продуктивність важлива тому, що без неї гра не досягне необхідної якості для кінцевого користувача. Нікому не потрібна гра, яка відображає низьку кількість кадрів в секунду, або некоректно працює при великій кількості користувачів що грають онлайн.

В свою чергу, можливість розширення важлива тому, що це дає змогу витратити менше ресурсів розробки при введенні нових функціональних можливостей та модифікації існуючих. При правильному проєктуванні важливо приділяти увагу шляхам розширення функціональності. Прикладом цьому може стати введення нового типу взаємодії користувача з навколишнім світом.

Найважливішим моментом на етапі проєктування гри є визначення ієрархії та поверхневого плану взаємодії ігрових елементів. Цей процес був виділений в окремий патерн – ЕС (Entity-Component). Патерн ЕС буде досить корисним на початку створення проєкту, для розбиття програмних компонентів по директоріям та шарам.

Типовим завданням для ігор є створення менеджерів, область відповідальності яких – управління ігровими процесами в цілому. Такі менеджери повинні мати доступ до необхідних об'єктів щоб контролювати їх.

Але, для того, щоб уникнути великої кількості прямих зав'язків між сутностями – доцільно буде використати патерн Observer. Цей патерн дає змогу об'єктам підписуватись на події (events) менеджерів, та реагувати на них необхідним чином. Також цей патерн вирішує проблему тісного зв'язку між компонентами системи, що дає можливість вносити зміни лише в один компонент, замість багатьох.

У кожній грі є необхідність в реалізації інтерфейсу користувача. Для цієї типової задачі доцільно використати досить типовий патерн MVVM (Model-View-ViewModel). Патерн дає змогу розділити взаємодію користувача з системою на три шари з різними областями відповідальності: представлення даних, їх зберігання, та посередником.

Одним з основних патернів при розробці ігор з великою кількістю об'єктів є використання оптимізаційного патерну – Object Pool. Патерн призначений для оптимізації процесу створення та знищення ігрових об'єктів. Це досягається за допомогою створення тимчасового сховища для об'єктів які необхідно знищити або створити. Суть полягає в тому, що об'єкти ховаються та відображаються на екрані, замість справжнього знищення, яке потребує звернення до оперативної пам'яті.

Використання патернів безперечно має позитивний вплив на архітектуру, масштабованість та чистоту вихідного коду гри. Але необхідно пам'ятати, що надвелика кількість патернів може призвести до заплутаної, тісно-зв'язаної системи, яка насправді не потребує абстракцій в такій кількості. Саме тому необхідно вміти оцінювати потреби та ризики при використанні патернів.

Обґрунтовано використання патернів проєктування, як життєво-важливого елементу створення високоякісного програмного продукту, в даному випадку – гри, з можливістю розширюваності та подальшої підтримки.