

СИСТЕМА ВІЗУАЛЬНОГО СКРИПТУ ЯК ЗАСІБ ПРОГРАМУВАННЯ

Програмувати на блупрінтах зовсім не складно, адже немає необхідності вчити складний синтаксис і писати код, а візуальна взаємодія дуже допомагає орієнтуватися в алгоритмі і «писати» програми швидше.

Blueprint (англ. план, креслення) – система візуального скриптинга, що представляє собою візуальний інтерфейс для створення елементів Геймплей, використовувана в ігровому движку Unreal Engine [1,2]. Вся логіка мови Blueprint будується з нодів (або блоків), які з'єднуються проводами. Яким би складними або простим він не здавався, він залишається досить таки потужним інструментом, на якому можна створити майже що завгодно, від простенького персонажа або відкриття дверцят до процедурної генерації рівня.

Два найбільш часто використовуваних типи блюпрінтів, блюпрінти-рівнів і блюпрінти-класи. Ноди – це візуальні уявлення подій, функцій і змінних [1]. Вони мають колірний код, що виражає їх призначення. Червоний нод – це нод події, що використовується для ініціювання виконання послідовності нодов. Сині Ноди – це функції для виконання певних операцій. Кольорові овальні Ноди, кожен з яких має тільки по одному контакту даних, представляють змінні.

Можна уявити процес візуального програмування як роботу електричного кола. Червоний нод події відправляє сигнал, що йде по проводах і запускає виконання будь-якого нода, через який цей сигнал проходить. Коли нод отримує сигнал, він отримує дані, які йому потрібні в контактах даних в лівій частині нода. Після чого нод виконує свою операцію, відправляє сигнал далі і повертає результати через контакти даних в правій частині нода.

Контакти даних мають колірний код, який базується на типі використовуваних ними даних. Контакти даних в лівій частині нода витягають дані, в той час як контакти даних в правій частині нода повертають дані [2].

Провода з'єднують Ноди. Колір кожного проводу відображає тип використовуваних їм даних. Подія (event) – це те, що відбувається в процесі гри, від натискання гравцем клавіші на клавіатурі або попадання аватара в якесь приміщення до зіткнення актора з іншим актором або початку гри. Події використовуються для ініціювання послідовності в блюпрінті. При запуску події з виведення ехес-контакту надсилається сигнал, який проходить по дротах і обробляється всіма функціями, зустрінутими їм на шляху. Коли сигнал доходить до кінця послідовності нодів, він пропадає.

При запуску гри всі Blueprint коди переводяться на мову C++. У самій грі використовується вже перекладений код на C++. У професійного програміста різниця між скриптом на Blueprint і на C++ може бути майже непомітна.

Самі Epic Games рекомендують використовувати блюпрінти, коли в проєкті дуже багато посилань на контент, а його логіка працює в першу чергу на візуальну складову [2]. Також вони стануть в нагоді при створенні прототипів, прямолінійної або рідко використовуваної логіки, яка не є частиною основної архітектури. Все, що не отримує переваг в C++ з точки зору продуктивності, масштабованості і стабільності, також може бути створено в Blueprints.

Блюпрінти виграють у C++ на початкових етапах розробки, особливо якщо код гри пишеться з нуля. Вони не вимагають установки додаткової середовища, до того ж пропонують швидкі ітерації. А блоковий синтаксис блюпрінтів зрозумілий не тільки програмістам, але і тим, хто знайомий з аналогічними системами в програмах для створення контенту – наприклад, художникам.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Александр Буев. *Основные приёмы blueprint-программирования* — 2020. Режим доступа: <https://it-cube48.ru/archives/18002>

2. Арам Куксон, Райан Даулингсока, Клинтон Крамплер. *Разработка игр на Unreal Engine 4 за 24 часа; [перевод с английского М. А. Райтмана]*. — Москва : Эксмо, 2019. — 528 с.