

АНАЛІЗ МОВ ПРОГРАМУВАННЯ ДЛЯ НАПИСАННЯ ДОДАТКІВ ДО ВІДЕОІГОР

Коли розробники надають спільноті можливість модифікувати гру, вибір правильної мови програмування стає критично важливим. Від цього залежить не лише зручність створення модів, але й продуктивність гри, стабільність та доступність інструментів для розробників-ентузіастів.

Одними з найпоширеніших мов є C++ та Python [1]. C++ широко використовується в ігровій індустрії завдяки своїй високій продуктивності та можливостям управління пам'яттю, тоді як Python відомий своєю простотою, швидкістю розробки та активним ком'юніті [2]. Вибір між цими двома мовами впливає на те, наскільки ефективними, доступними та стабільними будуть моди. Щоб визначити, яка мова програмування краще підходить для створення модів, необхідно оцінити її за такими вимірюваними параметрами:

Довжина коду – порівнюється кількість рядків коду, необхідна для реалізації однієї й тієї ж функціональності. Менша довжина коду може прискорити розробку та зменшити кількість потенційних помилок.

Читабельність – наскільки легко розуміти та писати код. Високорівневі мови зазвичай мають більш зрозумілий синтаксис, що спрощує розробку модів навіть для новачків.

Швидкість виконання – аналізується час виконання програми, щоб визначити, яка мова швидше обробляє код і виконує команди. Це критично важливо для модів, які працюють у реальному часі та впливають на продуктивність гри.

Портативність – чи можна використовувати код на різних платформах (Windows, macOS, Linux, консолі). Чим краща кросплатформеність, тим ширше буде коло користувачів модифікацій.

Для того щоб визначити ці характеристики було написано напишемо просту програму.

Код програми в C++ та на мові Python було написано, відкомпільовано і запущено.

Швидкість виконання - з порівняння цих двох мов програмування в таблиці 1 видно, що C++ демонструє кращу швидкість виконання порівняно з Python [2]. Довжина коду - можна помітити, що програма на Python складається лише з 5 рядків коду, тоді як програма на C++ містить 16 рядків.

Портативність – C++ попередньо компілюється в машинний код, і для компіляції використовується компілятор. Це означає, що той самий код не

можна просто так запустити на різних машинах чи операційних системах [3]. Натомість Python використовує інтерпретатор для виконання коду, що дозволяє йому бути кросплатформним і виконуватися майже на будь-якій машині або операційній системі [3].

Таблиця 1 – Порівняння часу виконання програм на C++ та Python

№	Час виконання для програми на C++ (в секундах)	Час виконання для програми на Python (в секундах)
1	1.395392	1.490080
2	1.458812	1.528273
3	1.400649	1.410740
4	1.500360	1.714277
5	1.499676	1.585119
6	1.481390	1.678517
7	1.500241	1.877171
8	1.361943	1.446538
9	1.317074	1.515121
10	1.336213	1.599779
Середнє	1.425175	1.584561

Читабельність – загального спостереження можна сказати, що програма на Python є значно простішою для розуміння, оскільки її синтаксис більше схожий на людську мову [3]. В той час як C++ має чітко визначений синтаксис і структуру, що робить код на C++ більш складним для розуміння.

Незважаючи на те, що мова програмування C++ має вищу продуктивність та є швидшою в порівнянні з Python [2], останній все ж краще підходить для створення модів у відеоіграх. Python, будучи більш зрозумілим та зручним для новачків, дозволяє швидше писати та тестувати код. Крім того, програми, написані на Python, мають значно меншу кількість рядків, що робить їх легшими для підтримки та модифікації. Його інтерпретаторна природа дозволяє забезпечити високу портативність і можливість запуску на різних платформах. Тому, для широкої спільноти моддерів, зокрема для тих, хто не має досвіду в програмуванні, Python є оптимальним вибором.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Рейтинг мов програмування 2021. Режим доступу: <https://dou.ua/lenta/articles/language-rating-jan-2021/>
2. Comparative Analysis of C++ and Python in Terms of Memory and Time. Режим доступу: <https://www.preprints.org/manuscript/202012.0516/v1/download>
3. A Comparison between Python and C++. Режим доступу: <https://www.jetir.org/papers/JETIREQ06006.pdf>