

АНАЛІЗ ПІДХОДІВ ДО ОЦІНЮВАННЯ ЯКОСТІ ПРОГРАМНИХ ПРОДУКТІВ**Кіцула В.І.¹⁾, Терлецький А.І.²⁾, Манжула В.І.³⁾***Тернопільський національний економічний університет**^{1,2)} магістрант; ³⁾ к.т.н., доцент***I. Постановка проблеми**

Поняття «якість» є складним і багатогранним. Зазвичай під якістю розуміється відповідність об'єкта якимось пропонованим вимогам. В програмній інженерії цей термін можна трактувати по різному. Можна трактувати якість програмного продукту як відповідність його характеристикам, визначених у вимогах до продукту. Це означає, що кінцевий продукт вирішує поставлені перед ним завдання. Інша інтерпретація може полягати в тому, що якість продукту – це відсутність дефектів і помилок, пов'язаних з аварійною роботою продукту. Також можна трактувати поняття якості як здатність до легкої зміни програмного коду і можливість легкого додавання додаткових функціональних можливостей до продукту. Як видно, термін «якість» може використовуватися на різних рівнях і в різних контекстах. Тому необхідно розрізнити два головних поняття – «якість програмного продукту» і «якість програмного коду» [1].

Якість програмного продукту визначається тим, наскільки він вирішує завдання кінцевих користувачів. Серйозний вплив на якість продукту надає коректність поставлених вимог до продукту на етапі аналізу.

Якість програмного коду увазі грамотно витриманий архітектурний стиль програмного коду, чіткий поділ коду на функціональні блоки і сувору структурування.

Таким чином, якісний програмний код – це код, який легко підтримувати, вносити в нього додаткову функціональність, змінювати існуючі алгоритми. Для отримання якісного програмного коду на сьогоднішній день розробники користуються різними методологіями і практиками, застосовують шаблони проектування, для типових завдань використовують готові бібліотеки і алгоритми.

Видно суттєва різниця між поняттями «якість продукту» і «якість коду». Продукт може мати якісний код і при цьому не вирішувати проблем кінцевих користувачів. Такий програмний продукт вважається неякісним. І навпаки, якісний продукт, який успішно справляється з завданнями користувачів, може складатися з неякісного коду.

Мета кожної розроблюваної програмної системи – вирішити задачі користувача. Але практично кожна програмна система має тенденції до подальшого розвитку і модифікації. При цьому важливими показниками є кількість дефектів в програмі і вартість модифікації коду. Якщо при додаванні нової функціональності виникає неприйнятне число помилок, то продукт вже не може задовольняти потреб користувача. Аналогічно, якщо вартість додавання нової функціональності в програмний продукт занадто висока, то це також негативно позначається на користувачі.

З цього випливає висновок, що якість програмного коду – це не функціональний, але дуже важливий показник, який згодом впливає на якість кінцевого програмного продукту. Якість коду в свою чергу визначає ефективність роботи програміста, тому що якщо програміст пише якісний код, то знижується імовірність того, що в коді будуть виявлені помилки.

В результаті досліджень, проведених в університеті короля Абдель Азіза [2], було доведено, що метрики складності, наприклад такі, як кількість нащадків, кількість методів на клас, дійсно допомагають оцінити якість коду. Для того щоб контролювати якісні показники, необхідно мати їхні чисельні характеристики. Тобто, для контролю якості програмного коду необхідно вміти вимірювати цю якість. Для отримання формальних оцінок якості програмного забезпечення існують метрики програмного коду. Суть цього механізму полягає в тому, що на основі аналізу вихідного коду програмної системи можна отримати різні числові характеристики. Зазвичай обчислення таких показників будується на основі аналізу графа програмного потоку або структури програмного коду. На сьогоднішній день існує велика кількість метрик коду, що аналізують різні аспекти програмного коду. Перевага метрик програмного коду полягає в тому, що в процесі їх обчислення не бере участь людський фактор, а всі

обчислення проводить комп'ютер. Це гарантує факт точності і повторюваності таких вимірювань для кожної з метрик

Для побудови об'єктивного уявлення про програмний код можна використовувати набір метрик, які будуть відображати цілісне уявлення про якість програмного коду. Як правило, якісний код представляється як програмний код, що не наділений надлишковою складністю і зв'язністю частин системи, добре структурований і має адекватні пропорції для об'єму.

Висновки

Складність розроблюваних додатків зростає з кожним днем, отже, збільшуються витрати на реалізацію проекту. В таких умовах проблема аналізу якості коду та ефективності роботи програміста стоїть особливо гостро. Існуючі на сьогоднішній день інструменти не пропонують прості і зручні можливості оцінити стан проекту.

Тому розробка сервісів, що надають можливість оцінювати ефективність розробника і в цілому проводити аналіз якості коду є актуальною задачею.

Список використаних джерел

1. Kaner C. Software Engineering Metrics: What Do They Measure and How Do We Know? // 10th international software metrics symposium, metrics, 2004, P. 12- 24.
2. Моисеев М. Ю., Карпенко А. В. Расчет метрик надежности программ на основе статического анализа // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. – СПб : СПбГПУ, 2009. Вып. 80. С.139- 147.

УДК 004.4.273

ВІЗУАЛЬНЕ ПРОЕКТУВАННЯ ГРАФІЧНИХ ІГРОВИХ СЦЕН В РЕАЛЬНОМУ ЧАСІ

Кіцула О.П.

Тернопільський національний економічний університет, магістрант

I. Вступ

Сьогодні більшість ігрових проектів супроводжується власними редакторами ігрових сцен. Однак дуже часто вони є складними в освоєнні, і лише деякі з них надають механізм розширення базового набору об'єктів сцени. До того ж більшість ігрових редакторів для встановлення динамічних властивостей об'єктів використовує скриптові мови, вивчення яких доцільно лише для фахівців комп'ютерної графіки.

Таким чином, стає очевидним той факт, що сучасна конкурентоспроможна гра повинна супроводжуватися простим, інтуїтивно зрозумілим і в той же час функціональним редактором сцен. Для того, щоб створення нової сцени було доступно будь-якій людині, необхідно, не застосовуючи спеціалізованих мов, звести весь процес до візуального проектування. При цьому весь набір дій повинен обмежуватися знайомими будь-якому користувачу ПК операціями: виділення об'єкта, перетягування об'єкта, задання певних властивостей об'єкта. Користувач повинен бачити створенні ним зміни сцени в реальному часі. Тому можна припустити, що програма редактора сцени і програма відображення сцени повинні використовувати один і той же модуль візуалізації.

В даний час доступна безліч аналогів програм відтворення зображень моделей об'єктів (движків), як у вихідних кодах, так і у вигляді виконуваних файлах. При цьому користувач цих програм не має можливості на свій розсуд змінити склад представлених там об'єктів і форму рельєфу, тобто не може редагувати сцену. Професійні додатки малоефективні для вирішення локальних завдань. Вони виконуються на замовлення і, як правило, мають високу вартість.

II. Мета роботи

Метою роботи є створення редактора графічний ігрових сцен із візуальним проектуванням 3D об'єктів в реальному часі.