

УДК 004.925.8

## ПРОЄКТУВАННЯ ТА РОЗРОБКА GAME-READY 3D-МОДЕЛІ З ПОВНИМ ВИРОБНИЧИМ ЦИКЛОМ У СЕРЕДОВИЩІ BLENDER ТА UNREALENGINE

Гольдберг М.І., кандидат технічних наук, доцент  
*Київський національний університет технологій та дизайну*  
Гідзіль О.О., студент  
*Київський національний університет технологій та дизайну*

*Ключові слова:* 3D-моделювання, Blender, UnrealEngine 5, UML-моделювання, PBR-текстурування, FBX-експорт, запікання карт.

Створення сучасного ігрового контенту вимагає суворого дотримання технологічного конвеєра (pipeline). Game-ready модель — це не лише візуальний образ, а структурований набір даних, оптимізований для обробки графічним процесором (GPU) у реальному часі. Головною проблемою є пошук балансу між візуальною деталізацією та продуктивністю. Високополігональні моделі (high-poly) не можуть бути використані в рушії напряму через надмірне навантаження на GPU. Актуальність дослідження полягає у розробці та обґрунтуванні повного циклу трансформації даних від деталізованого скульпту до оптимізованого ігрового асета.

Виробничий цикл розглядається як модульна система обробки даних, де результат попереднього етапу є вхідною умовою для наступного. Процес декомпозовано на сім послідовних стадій, що представлені на рис. 1.

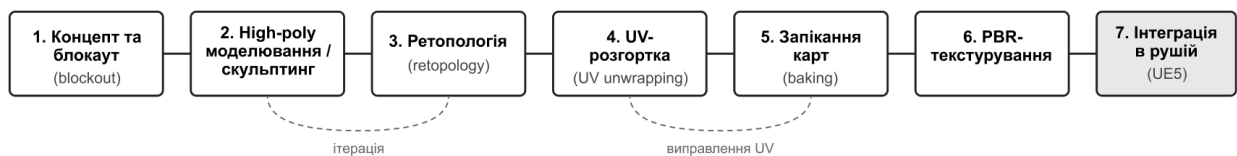


Рисунок 1 – Виробничий цикл game-ready 3D-моделі

1. Концепт та блокаут. Створення чорної геометрії з примітивів для перевірки силуету та масштабу.

2. High-poly моделювання. Будується деталізована сітка (мільйони полігонів) за допомогою скульптингу або модифікатора SubdivisionSurface. Це джерело інформації про дрібні деталі.

3. Ретопологія. Створення оптимізованої low-poly сітки поверх high-poly. Основна вимога — правильний напрямок ребер (edgeflow) для коректного затінення та збереження силуету при мінімальному полігональному бюджеті.

4. UV-розгортка. Перетворення 3D-поверхні у 2D-площину. Коректна розгортка вимагає рівномірної щільності текселів та відсутності перетинів (overlaps), що є критичним для наступного етапу.

5. Запікання карт (Baking). Перенесення деталей з high-poly на low-poly методом трасування променів (raycasting). Генеруються карти Normal, AmbientOcclusion, Curvature.

6. PBR-текстурування. Створення матеріалів на основі фізичних властивостей поверхні (шорсткість, металічність). Використовується модель PhysicallyBasedRendering.

7. Інтеграція в рушій. Експорт у форматі FBX, налаштування шейдерів в UnrealEngine 5 (UE5), генерація колізії та рівнів деталізації (LOD).

Для забезпечення стабільної роботи в UE5 модель має проходити технічну перевірку. Ключові критерії включають: відсутність N-кутників (примусова тріангуляція перед експортом для уникнення артефактів затінення); замкнену геометрію (Manifold) що передбачає відсутність внутрішніх граней та дірок у сітці; коректність нормалей коли усі вектори нормалей мають бути спрямовані назовні (враховуючи backfaceculling у рушіях).

Для забезпечення коректної інтеграції та узгодження систем координат між Blender (де вісь Z спрямована вгору) та UnrealEngine 5, конфігурація FBX-експортера має базуватися на ряді наступних критичних параметрів.

Обов'язковим є застосування трансформацій (ApplyTransform), що дозволяє зафіксувати масштаб об'єкта та уникнути помилок масштабування безпосередньо в ігровому рушії. Також, для передачі інформації про тверді та м'які ребра необхідно встановити параметр згладжування (Smoothing) що гарантує коректне відображення.

Важливим аспектом є налаштування орієнтації осей: для правильного позиціонування моделі в просторі UE5 встановлюються параметри -YForward та ZUp. Крім того, для забезпечення ідентичності геометрії в обох середовищах та запобігання появі артефактів при автоматичній інтерпретації полігонів рушієм, застосовується примусова тріангуляція (Triangulate) під час експорту.

Проектування виробничого циклу дозволило сформувати інженерну базу для створення якісного ігрового контенту. Використання Blender як єдиного середовища розробки забезпечує відсутність втрат даних при конвертації, а інтеграція в UE5 підтверджує сумісність розробленої моделі з сучасними галузевими стандартами. Запропонована трирівнева архітектура (Source → Processing → Engine) є універсальною та може бути застосована для розробки асетів будь-якої складності.

#### Список використаних джерел

1. Hristov G., Kyuchukova D. A Workflow for Developing Game Assets for Video Games. 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, 2021. DOI: <https://doi.org/10.1109/HORA52670.2021.9461355>.

2. FBX Static Mesh Pipeline. Epic Games. Unreal Engine Documentation. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/fbx-static-mesh-pipeline-in-unreal-engine> (дата звернення: 20.04.2026).