

THE PROCESS OF CREATING 3D MODELS IN BLENDER AND THEIR INTEGRATION INTO UNREAL ENGINE FOR GAME APPLICATIONS

¹Вінницький національний технічний університет

Анотація

У роботі представлено процес створення 3D-моделей у Blender та їх інтеграції в Unreal Engine для використання в ігрових застосунках. Проаналізовано етапи виробництва 3D-контенту, зокрема моделювання, UV-розгортку, текстурування та оптимізацію. Запропоновано практичний pipeline експорту ассетів і їх імпорту в Unreal Engine з урахуванням вимог до продуктивності в режимі реального часу. Описано структуру ігрових сцен, налаштування матеріалів та використання технік рівнів деталізації. Експериментальне тестування підтвердило коректність інтеграції ассетів і стабільну продуктивність у середовищі реального часу.

Ключові слова: 3D-моделювання, Blender, Unreal Engine, ігрові застосунки, рендеринг у реальному часі.

Abstract

This paper presents the process of creating 3D models in Blender and their integration into Unreal Engine for use in game applications. The stages of 3D content production, including modeling, UV mapping, texturing, and optimization, are analyzed. A practical pipeline for exporting assets and importing them into Unreal Engine is proposed, taking into account performance requirements for real-time rendering. The structure of game scenes, material setup, and level-of-detail techniques are described. Experimental testing confirmed the correctness of asset integration and stable performance in a real-time environment.

Keywords: 3D modeling, Blender, Unreal Engine, game applications, real-time rendering.

Introduction

The modern computer game industry is characterized by rapid development of 3D graphics technologies and increasing demands for visual quality and performance of game applications. Realistic scenes, detailed models, and accurate lighting are key factors that influence the user's perception of a game. In this context, the use of effective tools for creating and integrating 3D content into real-time environments becomes especially relevant.

One of the most widely used tools for creating three-dimensional models is Blender, a free and open-source software that supports the full 3D production cycle, including modeling, texturing, and optimization. For implementing game logic and real-time rendering, Unreal Engine is commonly used, as it provides high performance, an advanced material system, and powerful scene optimization tools.

However, the process of integrating 3D models from a graphics editor into a game engine requires compliance with specific technical requirements. Incorrect scale settings, transformations, or material configurations can lead to rendering issues and reduced performance. Therefore, the analysis and formalization of the process of creating 3D models in Blender[1] and their correct integration into Unreal Engine[2] is a relevant research task.

The purpose of this study is to analyze and practically implement the process of creating 3D models in Blender and integrating them into Unreal Engine for use in game applications, taking into account real-time performance requirements.

Research Results

During the research, a practical pipeline[3] for creating 3D content for game applications was developed and implemented, covering all main stages from modeling to real-time usage. At the initial stage, the requirements for 3D models in game engines were analyzed, including limitations on polygon count, texture usage, and the impact of geometric complexity on performance.

The process of model creation in Blender included polygonal modeling with a focus on a low-poly approach, UV unwrapping[4], and texturing using PBR materials[5]. To improve visual quality without increasing geometric complexity, normal maps and roughness maps were baked from high-poly models. Special attention was paid to topology optimization and the removal of unnecessary elements that could negatively affect performance.

Before exporting the models, scene preparation was performed in Blender, including applying transformations and correctly configuring scale and coordinate axes. The FBX format was used for data transfer between environments, as it ensures compatibility of geometry, materials[6], and UV coordinates.

The integration of models into Unreal Engine involved importing geometry, configuring materials, and connecting textures to the corresponding shader parameters. To improve scene performance, a level-of-detail (LOD)[7] system was implemented, allowing automatic reduction of model complexity depending on the distance from the camera. The created 3D assets were used in a game scene with basic lighting, and their correct real-time rendering was verified.

Testing of the implemented prototype was carried out by analyzing scene stability and measuring performance[8] under different object counts. The obtained results confirmed that the optimized models are correctly integrated into Unreal Engine and provide a stable frame rate without loss of visual quality.

Thus, the research demonstrated the effectiveness of using Blender in combination with Unreal Engine for creating and integrating 3D content into game applications while meeting real-time performance requirements.

Conclusions

As a result of the study, the process of creating 3D models in Blender and integrating them into Unreal Engine for game applications was analyzed and implemented. The main stages of the 3D pipeline were considered, including modeling, texturing, optimization, and the use of level-of-detail techniques. Practical testing confirmed correct asset integration and stable scene performance in real-time mode. The proposed approach can be applied in educational and indie game projects and serves as a basis for further functional expansion and increased scene complexity.

REFERENCES

1. Blender Foundation. Blender Documentation [Electronic resource] // Available at: <https://docs.blender.org> ; Accessed: 20.12.2025.
2. Epic Games, Inc. Unreal Engine Documentation [Electronic resource] // Available at: <https://docs.unrealengine.com> ; Accessed: 20.12.2025.
3. Epic Games, Inc. FBX Content Pipeline [Electronic resource] // Available at: <https://docs.unrealengine.com/5.0/en-US/fbx-content-pipeline-in-unreal-engine/> ; Accessed: 20.12.2025.
4. Blender Foundation. UV Unwrapping [Electronic resource] // Available at: <https://docs.blender.org/manual/en/latest/modeling/meshes/uv/unwrapping/index.html> ; Accessed: 21.12.2025.
5. Allegorithmic. Physically Based Rendering (PBR) Guide [Electronic resource] // Available at: <https://substance3d.adobe.com/documentation/> ; Accessed: 21.12.2025.
6. Epic Games, Inc. Materials and Shaders in Unreal Engine [Electronic resource] // Available at: <https://docs.unrealengine.com/5.0/en-US/materials-in-unreal-engine/>; Accessed: 21.12.2025.
7. Epic Games, Inc. Level of Detail (LOD) System [Electronic resource] // Available at: <https://docs.unrealengine.com/5.0/en-US/level-of-detail-in-unreal-engine/> ; Accessed: 20.12.2025.
8. Epic Games, Inc. Performance and Optimization [Electronic resource] // Available at: <https://docs.unrealengine.com/5.0/en-US/testing-and-optimizing-your-content/>; Accessed: 21.12.2025.

Нілова Марія Геннадіївна – студентка групи ІКН-25м, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, email:mariaaail.2231@gmail.com.

Мельник Олеся Дмитрівна — к.ф.н., доцент кафедри іноземних мов, Вінницький національний технічний університет, e-mail: prysyazna_lesya@vntu.edu.ua

Nilova Mariia Hennadiivna – student of Group 1KN-25m, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, Ukraine; e-mail: mariaanil.2231@gmail.com.

Melnyk Olesia D. — Candidate of Philological Sciences, Assistant Professor of the Department of Foreign Languages, Vinnytsia National Technical University, Vinnytsia, e-mail: prysyazna_lesya@vntu.edu.ua