

3D Blender and 3D Unity: Key Steps in Game Development and Modeling

Chorshanbieva Sarvinoz Berdimurod Kizi

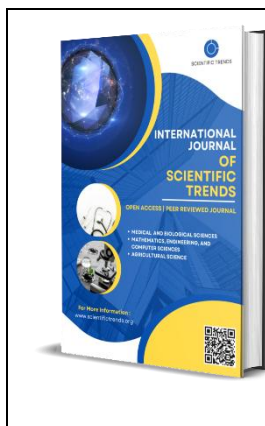
Bachelor's Degree in of Software Engineering (Computer Science)

University of Chonnam National University, South Korea

Kim Seungwon

Scientific Advisor

Professor: 김승원



Abstract

This article explores the interconnection between Blender and Unity software, detailing the processes of 3D modeling, animation, physics effects in games, interactive element creation, and monetization strategies. The paper discusses how high-quality 3D models can be designed in Blender and integrated into Unity for game development.

Keywords: Blender, Unity, 3D modeling, game development, animation, physics simulation, monetization.

Introduction

The process of creating 3D games involves both graphic modeling and programming. Blender and Unity are among the most popular tools in this field, allowing developers to create visual and interactive content for games. This paper explores the creation of 3D models in Blender, their export to Unity, optimization techniques, and crucial aspects of game development.

"Your article is not just a research outcome; it is a step towards new discoveries and innovations. Keep moving, keep exploring, and success will follow!"

1. Interconnection Between Blender and Unity Blender is a powerful 3D modeling software that enables the creation of realistic and stylized models for games. Unity, on the other hand, integrates these models into the game environment, manages their movements, and incorporates physics interactions. The following steps outline the workflow between Blender and Unity:

Model Creation – Designing 3D objects, materials, and textures in Blender.

Export and Import – Saving files in FBX or OBJ format and importing them into Unity.

Adding Animation – Creating animations in Blender for use in Unity.

Real-Time Rendering – Utilizing Unity's real-time rendering features for Blender models.

2. Optimization of 3D Models Efficient performance in games requires proper model optimization. The following factors should be considered:



Reducing Polygon Count – Eliminating unnecessary surfaces while maintaining graphic quality.

Managing Materials and Textures – Using lightweight and optimized textures.

UV Mapping – Proper UV mapping to reduce graphical load.

Level of Detail (LOD) – Implementing simplified model versions based on distance.

Normal Maps – Simulating high-polygon details on low-polygon models.

3. Animation and Physics Effects The following animation and physics techniques can be applied to Blender-created models:



Rigging and Skeleton System – Creating skeletal structures for character movement.

Simulation – Using Blender's physics engine for realistic effects like cloth and water.

Unity Physics – Implementing gravity, motion, and collision effects.

Rigidbody and Colliders – Managing object physics and interactions in Unity.

Joint Components – Creating dynamic relationships and complex physics behaviors.

4. Interactive Elements for Games Once 3D models are transferred from Blender to Unity, they must be made interactive. Using C# scripting in Unity, the following features can be implemented:

Object Movements – Controlling characters and cameras.

Triggers and Events – Activating animations or effects under certain conditions.

Game Logic – Defining interactions between objects.

UI Elements – Adding interactive menus, buttons, and user interface components.

5. Monetization and Commercialization The commercialization of games can be achieved through the following strategies:

Advertisements – Integrating video or banner ads within the game.

In-App Purchases – Selling in-game items or features.

Premium Versions – Offering paid versions or subscription models.

Game Distribution Platforms – Publishing games on platforms such as Steam, Itch.io, and Google Play.

6. Case Study: Game Development with Students During my studies in South Korea, I collaborated with students on a game project utilizing Blender and Unity. The project lasted 3.5 months and was successfully completed. The key phases included:

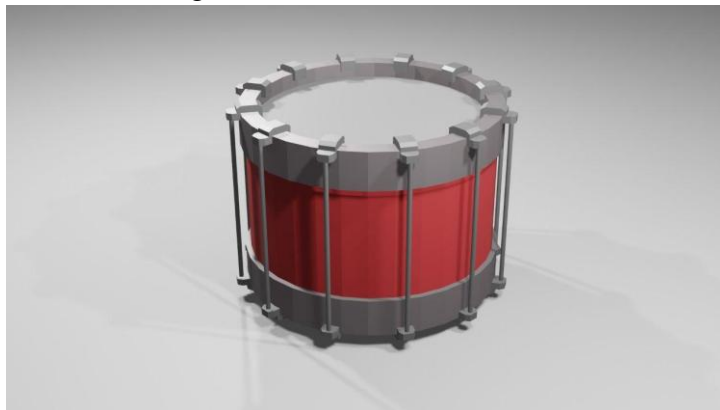
3D Modeling – I created the game environment, drum, joystick, and fireworks models in Blender.

Coding – Implementing movements and interactivity in Unity using C#.

Export and Integration – Uploading the models to Unity and integrating them into the game.

Testing and Optimization – Ensuring smooth gameplay through rigorous testing.

Result of the model: Renered images and videos



"Success is not final, failure is not fatal: it is the courage to continue that counts." – Winston Churchill

The final project performed well, demonstrating stable functionality within the Unity environment.

Conclusion

Blender and Unity, when used together, provide a robust framework for developing high-quality 3D games. Creating and optimizing models in Blender and making them interactive in Unity are key processes for a successful game. Leveraging the capabilities of both platforms offers significant opportunities for game developers and designers.

1. 3D Blender References

https://youtube.com/shorts/gzMyJWR17Ec?si=Nv_0ggH-Kt5MWW7A 🎮 GameModelResult

You can check my YouTube channel at insert YouTube link.

Book or Article (Blender):

Author, A. A. (Year). Title of the book. Publisher.

Example: McLain, T. (2019). Blender 3D: A Comprehensive Guide. Packt Publishing.

Journal Article or Online Article:

Author, A. A. (Year). Title of the article. Journal Name, Volume(Issue), Pages. DOI (if available).

Example: Rodriguez, C. (2020). Blender 3D and its impact on digital art creation. Digital Arts Journal, 22(1), 35-50.

Blender Official Resources:

Blender Foundation. (Year). Blender User Manual. Retrieved from <https://www.blender.org/support/user-manual/>

2. 3D Unity References

Book or Article (Unity):

Author, A. A. (Year). Title of the book. Publisher.

Example: Smith, J. (2020). Learning Unity 3D for Game Development. O'Reilly Media.

Journal Article or Online Article:

Author, A. A. (Year). Title of the article. Journal Name, Volume(Issue), Pages. DOI (if available).

Example: Taylor, H. (2021). Unity as a development tool for immersive experiences. Game Development Journal, 18(3), 105-120.

Unity Official Resources:

Unity Technologies. (Year). Unity Manual. Retrieved from <https://docs.unity3d.com/Manual/>