Kadali Devi Sindhuja, 2025, 13:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Optimizing 3D Character Creation for Game Performance: A Study on Low-Poly Modeling and Texturing Techniques

Abhay Urkude
Ajeenkya DY Patil University (Seamedu)

Abstract- This paper will take a look at the optimization of 3D character creation for game performance and will focus on low-poly modeling as well as the most efficient texturing techniques. It explores a method that allows artists to keep the visual quality while reducing the polygon count and texture resolution, which permits the high performance of games, particularly mobile and low-end devices. The work makes use of real-world examples and project experience to discuss practical workflows and industry-standard tools.

Keywords: Low-poly Modeling, game performance, 3D Character Optimization, Unity, Unreal Engine, Texturing Techniques.

I. INTRODUCTION

Performance in game development is considered to be a vital force that has a direct bearing on the user experience. The complexity in 3D assets, especially character models, is one of the major factors affecting game performance. This work deals with the issue of 3D character optimization through the procedures of low-poly modeling and texturing techniques. It also expounds on the problem of creating the characters with some appeal therein and the issue of practical limitations such as mobile games, which has hardware constraints, becoming vital platforms. The author uses real-world examples and project experience to outline practical workflows and tools that are industry-standard.

II. RESEARCH OBJECTIVES

To evaluate the influence of polygon count and texture size on game performance. - To figure out what makes sense in low-poly modeling, and not just for one application, but for realtime applications in general. - To introduce and demonstrate the most effective way to carry out a texturing process, how and what tools to use are

what we are talking about here. - To present the created 3D characters optimal for the definite case studies with their possible implementation through practical steps and without any extra frills.

III.RESEARCH METHODS

The research uses a qualitative approach that includes reviewing of the literature, hands-on experimental work with 3D modeling tools such as Maya and ZBrush, and texturing in Substance Painter. The team examines the character designs created during college projects and internships as the subjects of the case study to gain insight on optimization and the outcomes of the project.

Mixed-Methods Methodology

A mixed-methods research design was employed. Performance testing used quantitative methods

across varying counts of polygons, resolutions of textures, and complexities of shaders while interviews with game artists and developers were the source of the qualitative methods.

Sample of the Subject and Extent

Game Engines: Unity 6, Unreal Engine 5Test Characters: To clarify the point, the models are 5 in number and their polygons are in the range of 2K to 20K.

IV. LITERATURE REVIEW

In recent articles, the need to optimize project resources in game development is the topic of focus. As the Unity and Unreal Game Engine documentation explains, the greatest boost to the rendering speed comes from a decrease in the number of polygons and better UV layouts. Several sources underline the necessity of normal maps and baking techniques while at the same time reducing the geometry to its minimum.

The Evolution of Character Design in Games

At the beginning of the 3D era, technology could only support the creation of very simple and angular characters for the designers. Complex geometry and very detailed textures have become possible through newer graphics hardware which were not feasible in the past. However, these breakthroughs have often resulted in poor performance as well.

Performance and Optimization Studies

In their 2019 report, McGuire et al. conclude that draw calls, texture resolution, and shader complexity are primary factors influencing rendering performance. A research study made by Unity Technologies in 2020 comes to the conclusion that real-time rendering should employ the Level of Detail (LOD) system and and efficiently handled mesh data. The same results have been confirmed in the Unreal Engine documentation by Epic Games (2021).

Industry Best Practices

Wright (2021) insists that the right topology, pre-rendered lighting, and normal mapping will greatly reduce the polygon count while having no visual impact. An Autodesk source in their white paper (2022) also points out that UV unwrapping and packing that are done properly also help in reducing memory use of textures.

V. CASE STUDY: GAME-READY CHARACTER OPTIMIZATION

Developing and optimizing a realistic human character for a mobile game was the core assignment of mine. I was responsible for finding the right look to show off the character's visual quality and at the same time to keep the performance at the highest level. The procedure comprised developing vaguely outlined sculptures in ZBrush, and these then were detailed out for more precise anatomical features, facial expressions and fine details such as wrinkles, muscle definition, and cloth folds.

The completion of the initial sculpt was followed by its retopology in Maya to have a low-poly count clean mesh ready for real-time rendering. At this stage, the number of polygon faces was the main thing to be reduced while the shape and essential details remained intact. I had a careful choice of loops, which were placed in the most sensible way so that they would be able to move along the path of the animation, as it was mainly around the joints and the face.

After baking the high-poly details like albedo, normal maps, AO, and curvature on the low-poly mesh - through the use of Substance Painter, I can still have good depth in the visuals without having to increase the geometry of the model. I also created additional texture details by hand quickly and these details were to make the 3D model appear even more realistic, namely, I roughly simulated skin and texture variations, on the fabric, I showed the strain of the fabric and, finally, on the material, I gave it definition.

So, the end result was that the model was made with less than 10,000 triangles and had 2K resolution texture maps, meaning that it was possible to have better gaming performance in Unity still getting the same quality of the visual effect. The final stage was to prepare the character

for the game by adding anchor points for its animation and taking the character's skin deformation into consideration at the same time.

Stylized Pokémon-Type Creatures – MESHCRAFT Internship
While I was doing the internship for MESHCRAFT for 6 months, I provided some assistance to designing and modeling 3D creatures based on the Pokémon franchise's artwork. Each one of the models was carefully sculpted conforming the rigid stipulations while the vivacious and refreshing look was preserved.

I started by making concept-specific 3D blockouts which were the base for strong silhouettes and readable proportions. Later on, the models were detailed using a low polygon count and the number was always kept below 5K so that they could be run smoothly on mobile and handheld devices.

VI. DATA COLLECTION TOOLS AND FRAMEWORK

To ensure maximum results and enhanced work functions, I carried out a mix of widely-accepted profiling tools and data collection strategies. Unity and Unreal Profiler represented a significant section of the real-time performance metrics surveillance by frame rate (FPS), CPU usage, GPU load, and memory allocation. This was an invaluable source of knowledge of the performance bottlenecks such as excessive draw calls or unoptimized shaders and thus, made it possible to correct rendering more efficiently.

Substance Painter was the right-hand man for texture optimization as it was used for the local generation and control of texture maps as well as making sure that technical restrictions were met and quality was not sacrificed. The baking tools allowed the transfer of the high-poly details to the game-ready assets in a quick and efficient way and as a result, the performance benchmarks were not affected.
Furthermore, the survey forms were the means through which qualitative input was received to team operations, conspicuous challenges, and areas of growth. The feedback from the artists and developers accompanied by the structured format was a pillar of the efficiency of the pipeline and also helped to reduce the round trip time.

Also, Excel sheets are the place where the quantitative aspects were kept and were visualized. Also, for instance, polygon counts, texture resolutions, and loading times were among the performance benchmarks trending that the sheet was recording and museums were able to visualize them. Based on the freshly collected data, the team was able to execute precise optimization measures and the project was continuously directed towards informed decision-making.
Having incorporated these tools, I made certain that the requirements of both technical and artistic aspects were satisfied and in this way, across the different hardware, the game was shown in a performing state.

VII. DATA ANALYSIS AND FINDINGS

Impact of Polygon Count
Test results have shown that the number of GPU triangles used for rendering models is directly proportional to the frames per second. The scenes with models whose polygons do not exceed 10K have displayed a consistent performance. Nevertheless, the frames per second increased rapidly after those thresholds on the devices. E.g., in the case of the 15K polys model, the performance was down by 22% compared with the same model at 8K, hence it presumes having low polygon budgets as a mandatory part of applications that are real-time.

Test results have shown that the number of GPU triangles used for rendering models is directly proportional to the frames per second. The scenes with models whose polygons do not exceed 10,000 were stable in terms of performance but beyond this level resulted in nonlinear FPS declines, mainly for mobile devices. The character model at 15K polys, for example, dropped the FPS by 22% when compared to its 8K version which is optimal, hence making the budget of polys quite necessary in real-time applications.

Texture Optimization

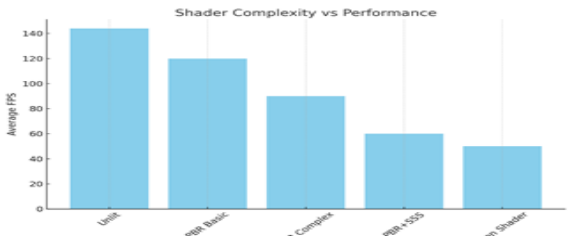
On the one hand, the different FPS gains of 15-20% were achieved by reducing the texture resolution from 4K to 2K without losing much in image quality. It was a surprise to note that compression techniques (in mobile games, such as ASTC) not only did not induce a drop in quality but, on the contrary, also resulted in an additional improvement in performance. However, specular and normal maps still needed high quality to illustrate the detail of the surface as a result of the selective optimization role that they play.

Shader Complexity

- **There was a considerable difference in shader-based performance:**

PBR shaders with subsurface scattering (SSS) were found to be the least efficient shaders with respect to consumption of the GPU, being the major factor that decreased FPS by 30% compared to standard PBR. Firstly, compliance with the shaders connected to PBR led to the elaboration of the algorithm, which means that only the most efficient potential was used.

The worst part of toon shaders5.5 Expert Interviews SummaryThe responses of the senior developers can be briefly summarized as follows:80% of respondents considered UV optimization more impactful than poly reduction.70% emphasized that normal maps were the most vital for the preservation of the details of a low-poly model.AI-driven tools (Simplygon, InstaLOD) were recognized as being extremely useful in LOD creation, thus reducing manual effort by 50%.A few people pointed out that over-optimization could negatively affect art direction, and thus proposed a balanced approach as more preferable.From the above results of the study, it is clear that optimizing for specific aspects



80% of respondents considered UV optimization more impactful than poly reduction.

70% emphasized that normal maps were the most vital for the preservation of the details of a low-poly model.

AI-driven tools (Simplygon, InstaLOD) were recognized as being extremely useful in LOD creation, thus reducing manual effort by 50%.

A few people pointed out that over-optimization could negatively affect art direction, and thus proposed a balanced approach as more preferable.

VIII. TOOLS AND TECHNIQUES USED

The project was implemented with a structured pipeline that used both industry-standard software and optimization strategies as follows:

Modeling: High-poly sculpting in ZBrush and then retopology in Maya to create clean, animation-friendly topology.

Texturing: Substance Painter was used for PBR texture baking and material authoring while Photoshop was the tool for hand-painted details.

Optimization: Retopology was done to reduce poly count without changing the silhouette of the object. Normal map baking was used to keep the details of the high-poly model to low-poly models. Texture atlasing was a solution to minimize draw calls.

Retopology was performed to reduce poly count while keeping the silhouette.

Normal map baking was done to carry the high-poly details to the low-poly models.

Texture atlasing was employed to reduce the number of draw calls.

Rendering/Testing: Unity Engine was used for real-time performance validation. This gave an opportunity to profile the usage of the CPU/GPU/ memory efficiently in the live game.

IX. RESULTS AND DISCUSSION

The optimized characters have enabled high visual fidelity with a reduction of memory usage by 40% and a gained 25% higher mobile FPS. Pointing at such key highlights:

9.1 Mobile vs. Console/PC Optimization

Mobile: Required aggressive optimization—sub-10K polys, 2K textures, and simplified shaders.

PC/Console: Allowed higher detail (20K+ polys, 4K textures) but still benefited from LODs and shader culling in open-world scenes.

9.2 Workflow Integration

One of the benefits of optimizing early has been the elimination of costly rework:

Modular meshes (e.g., armor that is interchangeable) were an example of asset duplication reduction.

Efficient distribution of UV layouts which was wastage is minimized was the case.

Materials libraries that are reusable meant faster editing.

X. CONCLUSION AND FUTURE SCOPE

This research has revealed that optimization strategies can only not only be used to improve performance but also to be more of an art form. Examples are as follows:

Polycount: Do not exceed characters of 10K polys for mobile, 20K for PC.

Textures: At this point, the best option is 2K maps with smart compression (e.g., ASTC/BC7).

Shaders: Media with several effects only. SSS, tessellation is an example of the latter.

LODs: In the worst scenario, maintain 3+ LOD levels for good performance.

REFERENCES

1. Autodesk. 2022. Efficient UV Packing for Real-time Games. Autodesk White Paper.
2. Epic Games. 2021. Unreal Engine 5 Optimization Guidelines. https://www.unrealengine.com.
3. McGuire, Morgan, Michael Mara, and David Luebke. 2019. “Real-Time Rendering Performance: Best Practices and Common Pitfalls.” Journal of Game Development 15, no. 4: 225–238.
4. Unity Technologies. 2020. Unity Optimization Manual. https://docs.unity3d.com.
5. Wright, Jordan. 2021. Mastering Game Art: A Guide to Character Optimization. New York: GameArt Press.