ARCHITECTURAL AND PROP MODELING FOR VIRTUAL REALITY: LOW-POLY TECHNIQUES IN 1980s RETROFUTURISTIC DESIGN

Uriel Haile Hernandez Belmonte, *professor in the Department of Art,*
University of Guanajuato, Mexico
Noel Adair Esteves Ruiz, *Digital Art student,*
University of Guanajuato, Mexico

Abstract. *This research details the low-poly techniques and manual optimization methodology used to create visually coherent, highly efficient 3D assets for the VR game Sentinella. Developed for the Oculus Go headset, which has limited processing power, the methodology focuses on manual topological optimization, including Edge Loop Removal and Vertex Merging, to maintain performance and visual quality. The workflow ensured stable frame rates and an immersive experience despite hardware limitations.*
Keywords: *virtual reality, low-poly, 3d assets, optimization.*

The creation of optimized 3D assets is a fundamental process in video game development, especially in virtual reality projects where hardware limitations directly affect performance and user experience. VR applications, unlike traditional video games, require stable frame rates and efficient rendering to maintain immersion and avoid issues such as latency, discomfort, or motion sickness. This optimization is crucial for the production pipeline. This research focuses on developing the VR game Sentinella for the Oculus Go virtual reality headset, a standalone device with limited processing power. Sentinella is a collaborative project between the University of Guanajuato and the Guanajuato Health Secretariat (SSG), designed to teach users to identify risk patterns. Due to the Oculus Go's technical limitations, development concentrated on creating low-poly assets and manual optimization techniques. The aim of this research is to detail the low-poly techniques and manual optimization methodology used to create visually coherent, highly efficient 3D assets for the Sentinella VR game, demonstrating the balance achieved among aesthetics, functionality, and performance despite the limitations of the Oculus Go headset.

The development of 3D assets in this project is based on a methodology focused on manual topological optimization and the application of strict geometric restrictions, with the objective of producing highly efficient models with a low polygon count that can be efficiently used by the Oculus Go virtual reality headset for the video game Sentinella. Unlike automated processes, this optimization is performed manually, allowing precise control over edge flow, silhouette preservation, and polygon distribution within each model.

The process begins with a basic blockout where simple shapes are used to define the main volumes of the objects. Afterwards, the topology is refined by applying different optimization methods to maintain a balance between visual quality and performance. This includes Edge Loop Removal, which is carried out in areas where there are no significant changes in the object’s shape and where additional geometry does not provide meaningful visual value. Removing unnecessary loops helps reduce the total polygon count and maintain a lighter and more efficient mesh. Vertex Merging is also used to reduce unnecessary density without compromising the overall shape of the model, which helps simplify the geometric structure and allows the asset to render more efficiently inside the game engine. Furthermore, Silhouette Control is prioritized: although the models use a reduced polygon count, maintaining a clear and recognizable silhouette is essential. In stylized and virtual reality video games, the visual readability of an object is more important than small geometric details, so the overall shape of the asset takes priority.

Each asset is developed under predefined polygon budgets according to its function within the video game. Props require minimal geometry, with emphasis on simple forms and implied detail. Main objects allow slightly higher density, but without exceeding overall limitations. Secondary and decorative objects use extremely simplified models to avoid unnecessary resource consumption. All geometry is designed for real-time engines such as Unity, prioritizing clean and optimized meshes. Non-visible faces are removed and geometry overlap is minimized to reduce rendering costs. Maintaining a stable performance rate was essential to avoid performance issues or user discomfort, making the optimization of each model a fundamental part of the workflow pipeline.

After completing a prop, the model is prepared for texturing through the UV mapping process. UV mapping is a fundamental concept in 3D modeling that plays a crucial role in the texture and appearance of models in computer graphics, functioning as a two-dimensional representation of the 3D model's surface that allows textures to be properly applied. In every model created for Sentinella, UVs are manually developed to maximize the available space within the UV map while avoiding texture distortion. An organized distribution of UV islands is also maintained to facilitate the later processes of texturing and editing.

Following the completion of the UVs, the model is exported in .fbx format. This format was selected because of its compatibility with different modeling software and video game engines, in addition to being relatively lightweight for file storage and transfer. The export process is carried out before texturing to maintain secure backups of the base model.

Due to optimization limitations and requirements, the details of a 3D model cannot be created solely through geometry, as this would consume resources unnecessarily. Therefore, much of the visual detail is added through Texturing. Using specialized software such as Adobe Substance 3D Painter, base colors are first applied to the model, and then details are added using digital painting tools and smart materials. This process simulates real-world materials and surfaces such as metal, wear, dirt, wood, or plastic, allowing the models to achieve greater visual richness without increasing the polygon count. Because of the game's cartoon-style artistic direction, textures are developed using stronger colors and stylized details instead of aiming for extreme realism. This helps maintain visual consistency throughout the project and contributes to making environments and objects easily recognizable to the player.

Once the models and textures are completed, the assets are imported into the game engine for Integration Within the Video Game. This stage involves performing performance, lighting, and scale testing within the virtual environment. It is verified that the models maintain good performance within the virtual reality headset and do not generate visual issues or performance drops. Adjustments to materials, collisions, and lighting settings are also made to ensure that each asset functions correctly within the video game Sentinella.

Thanks to this workflow methodology, it was possible to develop visually appealing and optimized assets while balancing visual quality, performance, and compatibility with virtual reality devices. In conclusion, the methodology used to create 3D assets for the video game Sentinella enabled the development of optimized, visually coherent models for implementation in virtual reality. Through manual topological optimization techniques, it was possible to maintain efficient performance despite the hardware limitations of the Oculus Go. Furthermore, the use of a cartoon-inspired aesthetic helped simplify geometry without losing visual identity, allowing environments and props to remain clear and visually appealing to the user. This entire process was necessary to ensure the video game ran properly on the Oculus Go virtual reality headset, meeting the device’s technical limitations while delivering a smooth, immersive experience.

References

1. Blender Deluxe. (n.d.). What are uv maps or uv mapping in 3d?.
2. Niixer. (2026). De la geometría al píxel: Dominando el uv mapping en blender 5.0.1 (moneda, dado y pirámide).
3. Noticias3D. (2018). Oculus go: Analysis and features.
4. RenderHub. (n. d.). 3d hard surface modeling techniques, workflows and industry insights.
5. SelfCAD. (2026). Hard-surface modeling for games – Techniques every 3d designer should know.