Rochester Institute of Technology

B. Thomas Golisano College of Computing and Information Sciences

Master of Science in Game Design and Development

Capstone Final Design & Development Approval Form

May XX, 2025

Student Name: **Richmond Matthew Hulse IV**

Research Title: **Measuring Efficiency Values of Node-Base geometry**

Keywords: **Node-Based, Time Efficiency, Blender, Houdini, Unreal 5.4**

Sten McKinzie
Lead Capstone Advisor

Professor David Sanchez
Research Advisor

David Schwartz, Ph.D.
Director, School of Interactive Games and Media

Measuring Value of Node Based Geometry

by Richmond Matthew Hulse IV

Paper submitted in partial fulfillment of the requirements for the degree of Master of Science in Game Design and Development

Rochester Institute of Technology

School of Interactive Games & Media

B. Thomas Golisano College of Computing and Information Sciences

May XX, 2025

Title

The first steps for everyone are: What is your problem, What is the significance, and what is the scope (ie write your future work first!). -- and does what you want to do fit in the structure below? It might not! That's okay. There are a lot of different ways to frame research.

The outline below covers the most common things you'll need to consider, but it won't be an exact match for everyone. You should work with Prof. Simkins & your committee on this!

Measuring Efficiency Value of Node-Based Geometry

Abstract

Set up a mini version of the whole thing.

This research evaluates the efficiency and adaptability of node-based geometry workflows compared to traditional modeling techniques in Blender 4.0, with a focus on their implementation at different stages of modeling. Node-based geometry, which combines modularity, precision, and scalability, offers an underexplored alternative to the emerging dominance of AI-driven asset generation. This study categorizes assets into simple, modular, and complex types to measure time efficiency, scalability, and iterative flexibility. The findings aim to demonstrate the unique advantages of Geometry Nodes, advocating for their broader adoption in game development pipelines and educational curricula.

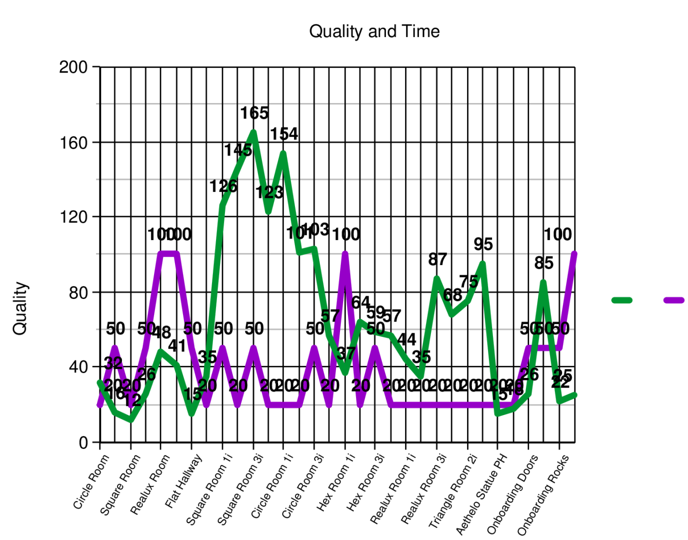
Problem Statement

What is the problem you're trying to solve? - in clear language that presents the problem in the context of prior work, the state of the field, and the perspective/context in which you'll be addressing this.

As game development pipelines become increasingly complex, the demand for efficient and adaptable asset production workflows has grown significantly. Traditional 3D modeling techniques, while versatile, often struggle to meet the iterative and scalable demands of modern projects. Meanwhile, AI-based asset generation has risen in popularity, offering speed and automation but frequently lacking consistency, adaptability, and creative control.

Amid these trends, node-based geometry workflows, such as Blender's Geometry Nodes, present a promising alternative. These systems combine modularity, precision, and scalability, enabling designers to make dynamic adjustments without disrupting the overall workflow. Despite their potential, node-based geometry remains underutilized in both industry and academia, with limited empirical studies evaluating their practical benefits in asset production.

This research seeks to address this gap by measuring the efficiency of node-based geometry workflows compared to traditional modeling methods. By examining their impact on asset creation across different stages of modeling—such as initial construction, detailing, and iterative modifications—this study will provide a comprehensive analysis of where and how node-based geometry can deliver value.



Significance

"So what?!" -- What is the significance of this problem? -- Why is this problem important to solve?

At the MS level, this may not be a novel work. Making a hypothesis and testing it in a context. Replication studies, application to a new context, collecting a variety of techniques, and comparing/contrasting within your context are all valid and worthwhile objectives.

The adoption of AI for asset generation in game development has surged, offering speed and automation in creating 3D models. However, this trend has often overshadowed the benefits of node-based geometry workflows, which provide a unique combination of modularity, precision, and control. Unlike AI-generated assets, which may lack consistency and adaptability for iterative design, node-based systems empower developers to dynamically adjust and fine-tune assets within a structured, non-destructive framework.

Node-based geometry, as exemplified in tools like Blender's Geometry Nodes, is fundamentally visual Python coding. This framework combines programming logic with visual design, making it an accessible yet powerful tool for both designers and technical artists. Despite its potential, the integration of geometry nodes into game development curricula, particularly in design- or programming-focused tracks, remains minimal. This absence is notable given the growing industry need for professionals who can bridge creative and technical workflows.

By exploring the efficiency and flexibility of node-based geometry, this research challenges the prevailing focus on AI, encouraging the field to reconsider the advantages of this established yet underutilized methodology. For game studios, particularly those balancing creative control with production efficiency, understanding the comparative benefits of node-based workflows can lead to more informed decisions and optimized asset pipelines. This study highlights the need for a balanced approach, showcasing how node-based geometry can complement or, in some cases, surpass AI-driven methods for asset creation in terms of scalability, precision, and integration into iterative design processes.

Background

Do you understand where your research is situated in the field? What have others done? You'll have some references in the definition of the problem. This section will go into more depth about related/similar work. The techniques, etc. you'll be building on. How is what you'll be doing comparable/different?

In other words, research doesn't come "out of thin air" - everything builds on the work of others. Did you do due diligence about the background of this problem and give credit where credit is due?

1. "*Graphical Modeling and Animation of Fracture*" by James F. O'Brien (2000)
   This doctoral thesis presents algorithms for simulating fracture in materials, foundational to tools like Digital Molecular Matter (DMM) used in games such as Star Wars: The Force Unleashed. The research highlights the importance of procedural methods in realistic asset destruction and deformation.
   + Link to thesis
     - **Author's Note**: James F. O'Brien is a professor at the University of California, Berkeley, renowned for his expertise in computer graphics and physical simulation. His research focuses on algorithms for simulating material behaviors, such as fracture and deformation, with applications in films, games, and

interactive tools like Digital Molecular Matter (DMM). O'Brien's work bridges academic innovation and industry application, significantly advancing physics-based animation.

2. "*Middleware Postmortem: IDV Inc S SpeedTreeRT*" (2004)
   This article examines the development of SpeedTree, a procedural generation tool for creating realistic vegetation in games and films, providing insights into early node-based methodologies.
   + Link to SpeedTree Info
     - **Author's Note**: Interactive Data Visualization, Inc. (IDV) is the company behind SpeedTree, a widely-used procedural modeling tool for creating realistic vegetation in games and films. Founded by Michael Sechrest and others, IDV has contributed to projects like Avatar, The Elder Scrolls, and Assassin's Creed. IDV's tools emphasize scalability and realism, setting a benchmark for procedural environmental assets.

3. "*Real-Time Deformation and Fracture in a Game Environment*" by Eric G. Parker and James F. O'Brien (2009)
   This paper discusses the implementation of real-time deformation and fracture algorithms within game environments, contributing to middleware like DMM. It explores how procedural techniques enhance realism in games.
   + Link to paper
     - **Author's Note**: Eric G. Parker, a computer graphics researcher, focuses on real-time simulations for interactive applications, including games. Collaborating with James F. O'Brien, he contributed to the development of deformation and fracture algorithms implemented in middleware like DMM. Together, their work enhances realism in game environments through procedural physics.

4. "*Node-Based Shape Grammar Representation and Editing*" (2013)
   This paper discusses a visual, node-based workflow for shape grammar representation, facilitating complex asset generation through recursive and parametric controls. It underscores the advantages of node-based systems in procedural content creation.
   + FDG 2013
     - **Author's Note**: This paper was presented at the Foundations of Digital Games (FDG) conference, a leading venue for research in procedural content generation, computational creativity, and game technology. FDG attracts top researchers exploring innovative techniques for scalable design and interactive storytelling in games.

5. "*Node-Based Particle System*" by Aitor Luque Bodet (2021)
   This study delves into the development of a node-based particle system, analyzing its integration within existing 3D software and its impact on asset creation workflows. The research highlights the flexibility and control offered by node-based systems in particle effects and asset production.

+ UPCommons
    - **Authors Note**: Aitor Luque Bodet completed his research at Universitat Politècnica de Catalunya (UPC) in Barcelona, a leading technical university specializing in engineering and computer science. His work focuses on integrating node-based systems into particle effects and asset production workflows, bridging academic research with practical software solutions for media production.

6. "*Blender Geometry Nodes: Node-Based Procedural Modelling for a Short Film*" by Mikko Lappi (2023)
   This bachelor's thesis evaluates Blender's Geometry Nodes in the context of short film production, focusing on procedural modeling's efficiency in creating complex environments. Lappi examines the benefits of procedural techniques over manual methods, particularly for large-scale scenes.
    + Theseus
        - **Author's Note**: Mikko Lappi conducted his research at Metropolia University of Applied Sciences in Finland, which emphasizes applied learning and industry collaboration. His bachelor's thesis highlights the efficiency and scalability of procedural modeling using Blender Geometry Nodes, particularly for creating complex environments in short film production.

7. "*Procedural Game Asset Creation with Geometry Nodes in Blender*" (2022)
   This article provides a practical guide on using Blender's Geometry Nodes for procedural asset creation, emphasizing data flow and node organization to enhance efficiency in game asset production.
    + Kodeco
        - **Authors Note:** Kodeco (formerly Raywenderlich.com) is a trusted online platform offering tutorials and guides for developers in programming and game development. Known for its industry-focused resources, Kodeco's content helps developers adopt tools like Blender Geometry Nodes to enhance efficiency in procedural game asset creation.

Methodology

What do you intend to do? State the hypothesis, introduce your taxonomy, set up the algorithm, etc. -- goals, big picture, what you'll do….

The structure here will vary depending on the type of research and what you'll need for each approach. You'll probably need some sub-sections.

- *Hypothesis, data to be collected, how you'll do analysis, dep, and ind variables, threats to validity…*
- *For a new approach, how will you assess that it worked? What evidence will you collect and what can you reasonably conclude from it?*
- *For classification research, clear identification of categories, rationale, related parameters, … - can't "prove" that taxonomy is correct, but you can use references, etc. to show it's relevant and appropriate to the problem in which you are applying it. Strengths and weaknesses of different approaches, etc. Again, it's not pulling it out of thin air!*

This research employs a comparative analysis of traditional modeling techniques and Geometry Nodes in Blender 4.0. The study focuses on evaluating the efficiency of both workflows using three distinct model types: simple models, modular models, and complex models. These categories capture a range of asset creation scenarios common in game development pipelines.

1. **Research Design**
   - The methodology involves recreating the same assets using both traditional modeling and Geometry Nodes. By categorizing assets into three types, the study examines how node-based workflows perform at varying levels of complexity and modularity:
     - Simple Models:
       + Basic assets such as chairs, barrels, or crates.
       + These models emphasize foundational geometric shapes with minimal detailing.
     - Modular Models:
       + Reusable assets like walls, stairs, or fences.
       + These assets are designed for replication and adaptation across larger scenes or levels.
     - Complex Models:
       + Intricate assets such as buildings, vehicles, or natural environments (e.g., trees).
       + These models include a mix of fine detailing and scalable components.

2. **Data Collection**
   - For each model type, the following steps will be taken:
     - Baseline Workflow (Traditional Modeling):
       + Create each asset using only traditional modeling techniques in Blender 4.0.
       + Record time-to-completion, iterations required for adjustments, and challenges faced.
     - Node-Based Workflow (Geometry Nodes):
       + Recreate a similar asset using Geometry Nodes/modifiers for various stages, including base modeling, detailing, and iterative modifications.
       + Measure the time taken and note any improvements in adaptability or modularity.

3. **Metrics for Evaluation**
   - To assess the effectiveness of Geometry Nodes compared to traditional workflows, the following metrics will be analyzed:
     - Time-to-Completion: The total time required to complete each model using both workflows.
     - Scalability: The ease of modifying or replicating assets, particularly for modular and complex models.
     - Adaptability: The ability to make iterative changes efficiently without rebuilding assets from scratch.
     - Detailing Complexity: The degree of control and precision achievable in fine detailing.

4. **Tools and Software**
   - Blender 4.0 will serve as the primary tool for both workflows. Task timers will be used to record time metrics, and detailed notes will be kept during each modeling session to document challenges and observations.

5. **Data Analysis**

- Quantitative data, such as time measurements, will be statistically analyzed to identify performance differences across workflows and model types. Qualitative feedback, such as ease of use and creative flexibility, will be thematically coded and evaluated.

6. **Comparative Framework**
   - The analysis will compare workflows across the three model types, focusing on:
     1. Efficiency at Different Complexity Levels: Highlighting where Geometry Nodes provide the most significant benefits.
     2. Impact of Node-Based Workflows on Modular Asset Design: Examining scalability and adaptability for reusable assets.
     3. Iterative Design Advantage: Assessing how Geometry Nodes support rapid adjustments and changes during development.

7. **Expected Outcomes**
   - This study is expected to demonstrate that Geometry Nodes significantly enhance efficiency, scalability, and iterative flexibility in modular and complex models. It will also identify specific scenarios where traditional workflows may still hold advantages, providing actionable insights for optimizing asset creation workflows.

Results

Like methodology, how you present results will vary depending on the type of research. For classification work, the two sections may blend a bit -- they'll be much more distinct for hypothesis -> test -> collect data -> ... approaches. The plans are in the previous sections, and the results and analysis are here.

Discussion

A more formal presentation of the conclusions you drew based on your results and analysis.

Conclusions

A wrap-up discussion. It may be merged with "Discussion" depending on the nature of your work. This would also include lessons learned about the research overall - methodology, analysis techniques, etc.

Future Work

This is where you define the line you aren't going to cross -- it defines your scope and what would be great, but beyond what you tackled for this research.

References

More on Citation:

Give credit to ideas that are not yours and the games that influenced you in ALL written work. Use APA style, which is standard in the Learning Sciences disciplines.

Recommendation: use Zotero free bibliographic software to automate your citations. It allows you to store citations for future reference and automatically reformats to different styles. Note that Zotero APA still requires capitalization correction to "sentence case" as opposed to "title case" - *https://www.zotero.org/support/google_docs*

Use the following page for learning in-text citations and how to write a references section: *https://owl.purdue.edu/owl/research_and_citation/apa_style/apa_style_introduction.html*

For games, cite designer (as author), year, and publisher using APA style – choose one of these:

- *http://askus.library.wwu.edu/faq/116850*
- *http://moonflowerdragon.blogspot.com/2010/03/how-to-cite-game-in-apa-style.html*
- *https://www.bibguru.com/g/apa-video-game-citation/*

If it is not your original idea, you must cite it!

1. O'Brien, J. F. (2000). Graphical modeling and animation of fracture (Doctoral dissertation, University of California, Berkeley). Retrieved from https://graphics.berkeley.edu/papers/OBrien-GMA-2000-08/

2. IDV Inc. (2004). Middleware postmortem: IDV Inc's SpeedTreeRT. SpeedTree. Retrieved from https://www.speedtree.com/about/
Gamasutra. (2004). SpeedTreeRT: A look behind the technology. Retrieved from https://www.gamasutra.com/view/news/228244/SpeedTreeRT_A_look_behind_the_technology.php

3. Parker, E. G., & O'Brien, J. F. (2009). Real-time deformation and fracture in a game environment. Retrieved from https://graphics.berkeley.edu/papers/Parker-RTD-2009-05/

4. Foundations of Digital Games (FDG). (2013). Node-based shape grammar representation and editing. FDG Conference Proceedings. Retrieved from https://dl.acm.org/conference/fdg

5. Bodet, A. L. (2021). Node-based particle system (Bachelor's thesis, Universitat Politècnica de Catalunya). Retrieved from https://upcommons.upc.edu/bitstream/handle/2117/394027/223_Memoria_TFG.pdf?sequence=2

6. Lappi, M. (2023). Blender geometry nodes: Node-based procedural modelling for a short film (Bachelor's thesis, Metropolia University of Applied Sciences). Retrieved from https://www.theseus.fi/handle/10024/808787

7. Kodeco. (2022). Procedural game asset creation with geometry nodes in Blender. Retrieved from https://www.kodeco.com/38674958-procedural-game-asset-creation-with-geometry-nodes-in-blender

8. Blender Foundation. (2023). Creating simple 3D assets in Blender. Blender Documentation. Retrieved from https://docs.blender.org/manual/en/latest/modeling/introduction.html
9. Blender Foundation. (2023). Designing modular assets for games. Blender Documentation. Retrieved from https://docs.blender.org/manual/en/latest/modeling/architecture.html
10. Blender Foundation. (2023). Advanced modeling techniques for complex 3D environments. Blender Documentation. Retrieved from https://docs.blender.org/manual/en/latest/modeling/advanced/index.html