

Computer Graphics Wizard Academy: Quest-based Learning to Engage Students Who Know Only High-School Geometry*

Sing Chun Lee¹

¹Department of Computer Science

Bucknell University

Lewisburg, PA 17837

singchun.lee@bucknell.edu

Abstract

At Bucknell University, a liberal arts college, many students majoring in computer science, art, and design encounter computer graphics without prior exposure to linear algebra and multivariable calculus, which can cause frustration and disengagement when faced with the mathematical concepts presented in a traditional graphics curriculum. To address this challenge in liberal arts colleges, I developed the *Computer Graphics Wizard Academy* - a quest-based course that introduces core graphic concepts, such as geometric transformations, rendering, and shading, using high school geometry as a foundation.

Grounded in experiential learning and narrative-driven learning, Computer Graphics Wizard Academy immerses students in a fantasy world where students take on the role of apprentice mages, master various computer graphics spells through scrolls and complete quests. Each scroll enables students to explore graphic concepts interactively and engage with the mathematics behind graphics visually. Each quest combines story elements with programming challenges and real-time WebGPU visual feedback, creating a context that sustains motivation and lowers

*Copyright ©2025 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

the barrier to abstract thinking. The open-ended nature of the difficulties invites students to personalize their solutions and promotes creative thinking.

In this paper, I present the course design and analyze student survey data to examine how narrative framing and creative affordances influence engagement, confidence, and learning outcomes. Our findings suggest that narrative-driven and hands-on learning environments can make computer graphics more accessible while also promoting deeper conceptual understanding and artistic experimentation.

1 Introduction

At Bucknell University, a liberal arts college, the computer science curriculum does not require linear algebra or multivariable calculus as a prerequisite. The absence of this mathematical foundation presents a significant challenge for computer graphics instructors and has led to low enrollment in graphics courses among students. Because of low enrollment, computer graphics have not been offered for several years. To counter this, I created Computer Graphics Wizard Academy (CGWA) — a quest-based course built around high school geometry and WebGPU. Inspired by Sheldon’s The Multiplayer Classroom [6], CGWA replaces lectures with interactive scrolls, assignments with fantasy quests, exams with trials, and the final project with a culminating final enchantment.

Unlike simplified tooling approaches, CGWA redefines graphics education by immersing students in a narrative where they cast graphics spells, write shaders, simulate transformations, and visualize geometry. Students use scrolls to study and interactively examine graphic concepts. They are tasked with completing quests that utilize the learned spells to create personalized solutions and experiment with them personally. To progress to the next level in this fantasy world, they need to pass the trials, which ensure their conceptual understanding of core graphic concepts. At the end of the course, the final enchantment showcase invites students to work together and synthesize their learning into a magical artifact of their own design, demonstrating both technical mastery and creative expression in a public celebration of their journey.

This paper describes the CGWA curriculum and explores its effectiveness through student surveys and project work. Our findings suggest that narrative-driven and hands-on environments, combined with creative autonomy, enhance accessibility and conceptual understanding in graphics education.

2 Related Work

The design of Computer Graphics Wizard Academy draws on research in shader-first graphics pedagogy, interactive visual feedback, and narrative-driven

learning. These studies collectively inform how CGWA uses interactive shader feedback, storytelling, and creative exploration to make core graphic concepts accessible and motivating for students without prior exposure to linear algebra or low-level graphics programming.

2.1 Shader-First Graphics Pedagogy

Modern graphics pedagogy has increasingly shifted toward shader programming, with shaders used as entry points for student exploration. This change is driven by pedagogical and technical factors: shader programming allows students to reason about transformations, shading, and interaction in a direct and visually observable manner. Talton and Fitzpatrick’s SIGCSE work on teaching with the OpenGL Shading Language emphasizes the pedagogical benefits of starting with shaders rather than fixed-function graphics pipelines [7]. Their course emphasized active experimentation with shader code - an approach that CGWA adopts through its scrolls, which allows students to see the effect of code edits in real time using WebGPU.

Fink *et al.* extended this approach through a software-based programmable renderer, showing that a shader-first pipeline improves conceptual comprehension of rendering and lighting operations without the complexity of full OpenGL environments [1]. Following along the same line, CGWA introduces graphic concepts, such as geometric transformations, shading, and physics-based simulation, through direct shader-level manipulation. However, it intentionally delays formal mathematical treatment, instead allowing intuition to emerge through visual feedback.

The emphasis on starting with visual feedback before introducing theory provides an essential precedent for CGWA’s experiential-first design: students learn by modifying shader code in a safe, exploratory context before formalizing these abstractions.

2.2 Visual Interaction and Conceptual Reinforcement

Even before the rise of shader-first teaching, tools like GraphicsMentor demonstrated the value of visual experimentation in conceptual learning. Nikolic and Shene showed that students who manipulated camera, light, and object parameters through a live interface developed stronger geometric intuition, even in the absence of explicit mathematical instruction [3]. By interacting directly with rendering parameters and seeing immediate results, students constructed mental models that supported later abstraction and understanding.

CGWA extends this principle. Each scroll is a minimal shader-powered sandbox where students can tinker with transformations, interpolation, or animation. For instance, a scroll on 2D projective geometric algebra might allow

learners to adjust bivector coefficients and observe the resulting object motions unfold without requiring them first to understand the geometric product. By the time formal abstractions are introduced in quests or trials, students have already internalized key ideas through repeated, manipulable exposure. This scaffolding approach allows students to engage deeply with concepts that would traditionally require a stronger mathematical foundation.

2.3 narrative-driven Learning and Motivation

While visual feedback builds intuition and motivation, sustained engagement often requires additional framing to maintain momentum. One strategy is to embed technical content within a compelling narrative. Sheldon’s The Multiplayer Classroom [6] formalizes this approach: students take on a character role, complete themed quests, and accumulate experience points as a substitute for grades. This structure transforms the classroom into a narrative-driven progression, where course content becomes part of the character development.

Sheldon’s framework has strong conceptual parallels with SIGCSE studies that show how storytelling and identity formation improve computing engagement. Storytelling Alice demonstrated that embedding programming into narrative-rich, character-driven environments significantly improved interest and self-efficacy, especially for students from underrepresented backgrounds [2]. Likewise, Parham-Mocello *et al.*’s Story Programming study emphasized that learners benefit when the storyline is introduced before coding; doing so helps frame the content as purposeful and coherent [4]. Fantasy-based framing has also been studied empirically. Scott and Ghinea [5] conducted a double-blind randomized controlled trial comparing a fantasy role-play debugging exercise to a standard control version. Their study found that students using the fantasy version showed statistically significant improvements in programming self-concept, suggesting that narrative framing can positively impact learner identity and motivation.

Although this work focused on procedural programming, the narrative structure and motivation model can inform CGWA’s application to graphics education directly. CGWA adopts the narrative structure fully: students play as apprentice mages, unlock graphics spells through scrolls and quests, level up by completing milestone trials, and finish it with a final enchantment.

3 Course Design

Computer Graphics Wizard Academy (CGWA)¹ is structured around a fantasy-inspired progression that replaces lectures with scrolls, assignments with themed

¹The course syllabus at <https://eg.bucknell.edu/~scl019/Courses/CGSP25/index.php>

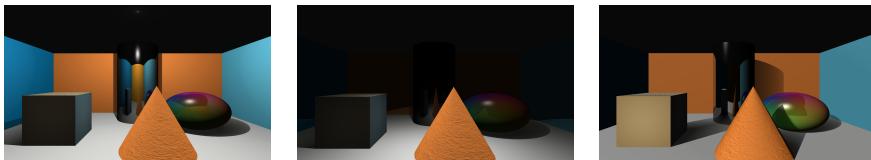


Figure 1: Students follow scroll instructions to modify shader code to change lighting implementation from point light (left) to spotlight (middle) and directional light (right).

quests, and assessments with performance-based trials. Students play the role of apprentice mages, learning “graphics spells” through scrolls (guided shader class exercises), completing quests (open-ended mini-projects), and culminating in a final enchantment (capstone showcase). The course is divided into three progressive phases, each designed to balance conceptual progression with increasing creative freedom.

3.1 Scrolls: Shader-Based Exploration and Visual Feedback

During the first two phases, students work through twenty-seven scrolls (interactive in-class exercises) that anchor abstract concepts in immediate visual feedback. Scrolls are structured to reflect best practices in shader-first pedagogy [7, 1]: students are introduced to a graphic concept through minimal narrative setup, brief theoretical framing, and incrementally editable shader code. Using WebGPU, students modify code and observe live visual outcomes in the browser.

This structure draws from work on exploratory visual learning [3]: rather than introducing formulas first, scrolls let students build intuition by manipulating effects directly. For instance, when learning 2D projective geometric algebra, students explore the impact of bivector coefficients on object motion before formalizing the geometric product. When studying lighting, students adjust WGSL shaders to implement point, spot, and directional lights (Fig. 1). These activities scaffold understanding while maintaining a strong sense of agency and play.

3.2 Quests: Open-Ended Application and Thematic Progression

Building on scroll foundations, weekly quests challenge students to integrate concepts in creative ways. Each quest is introduced within a fantasy scenario that aligns with recently learned scroll material. Unlike fixed-procedure labs, quests are open-ended: they ask students to solve problems using the “spells”

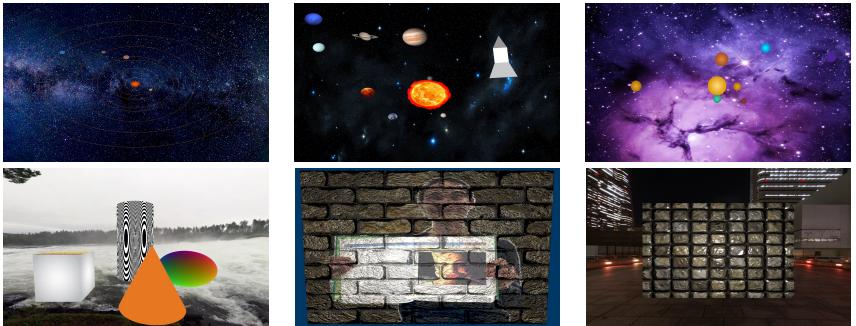


Figure 2: Upper: Left is the instructor’s demo of a 2D solar system; middle and right are students’ creative solutions to the same quest. Lower: Left is the demo of the environment and texture maps; middle and right are student quest implementations.

they have learned from the scrolls but leave space for visual experimentation, technical improvisation, and narrative invention.

The first phase includes five quests centered on 2D graphics and physical simulation. Students begin by rendering shapes using WebGPU (Quest 1), implementing animations using projective geometric algebra (Quest 2), and learning shading techniques (Quest 3). Quests 4 and 5 extend this into compute-based particle systems, spring simulations, and collision detection. These quests not only reinforce the shader-first model but also give students the creative autonomy discussed in [2, 4].

After completing their first trial, a performance-based checkpoint that evaluates conceptual understanding and shader fluency, students progress to the adept rank and begin phase two, which transitions into 3D graphics. This phase covers transformation matrices, camera projection, ray tracing, and procedural volume rendering. Quests 6–10 guide students through increasingly abstract levels while maintaining narrative framing. Student solutions vary widely (Fig. 2), reflecting the expressive potential of the format.

A second trial assesses mastery of ray-based rendering and geometric reasoning before students progress to mage rank and proceed to their final enchantment.

3.3 The Final Enchantment: Synthesis and Showcase

The final enchantment serves as both a capstone and a celebration. Working individually or in teams, students define a contract for a self-directed graphics project that must integrate multiple graphic techniques from the course. These

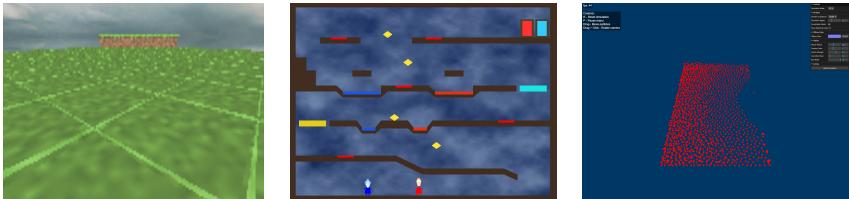


Figure 3: Students’ masterpieces: Minecraft VR (left), Fireboy and Watergirl (middle), and fluid simulation (right)

are presented as magical artifacts, and students are asked to design their own, estimate the effort required, and incorporate them into their narrative. Students are not simply proving proficiency, but they are authoring a final artifact that aligns technical fluency with creative expression [6, 5]. The showcase is public and celebratory, reinforcing learner identity and inviting peer recognition. Spring 2025 projects included a ray-traced Minecraft in VR, a WebGPU remake of Fireboy and Watergirl, and a real-time fluid simulation (Fig. 3), all developed from scratch using WebGPU, code, and skills that students learned from scrolls and quests ².

By placing narrative framing, shader-based experimentation, and open-ended creation at the core of its design, CGWA demonstrates that computer graphics can be taught in a way that is both accessible and rigorous, inviting creative expression while supporting conceptual learning.

4 Course Evaluation and Student Feedback

To explore how CGWA influenced student engagement, confidence, and conceptual understanding, particularly through its use of narrative framing and creative autonomy, I administered three structured surveys throughout the semester. Two custom-designed instruments were distributed after each mid-semester trial, and a summative course evaluation was collected via the university’s official system in the final week.

The mid-semester surveys³ combined 5-point Likert items, multiple-choice questions, and open-ended prompts aligned with our research themes: narrative immersion, motivation, conceptual clarity, and perceived effectiveness of the quest-based structure. The final evaluation added course-specific items to the institutional feedback framework.

²See demonstration here: <https://bucknell-graphics.github.io/>

³Survey forms can be found here: <https://forms.gle/1r8oyCUDYVFJkGs67> and <https://forms.gle/vXG6TBdvAnPYdtLq9>

4.1 Mid-Semester Survey 1: 2D Graphics Phase

The first survey, administered to 15 of 17 students (88% response rate) after the first trial, focused on the introductory 2D graphics. Results indicated high engagement with both the quest model and narrative structure. 80% of students reported enjoying the narrative format “a lot” or “a great deal” (ratings of 4 or 5), 80% found themselves feeling immersed in the fantasy narrative (ratings of 3 or above), and 73% agreed that the story elements helped them think more deeply about graphic concepts (ratings of 3 or above). Similarly, 87% stated that the quest format increased their motivation compared to traditional assignments (ratings of 3 or above), and 93% agreed that the quest-based format helped them effectively understand graphic concepts.

Only one of 17 preferred the traditional format, while more than half preferred the narrative-driven approach, with the rest opting for a mix of both. Students credited the narrative-driven model with more enjoyable coursework (80%), a sense of progression through the course (60%), and more creative problem-solving opportunities (60%). However, more than half (60%) found the narrative approach to be confusing due to its use of unfamiliar terminology and analogies. One student wrote, ‘I really like the quest format of having a single larger assignment per week.’ Another noted, ‘I think using the narrative format for abstract things is fine; calling daily class notes “scrolls” and weekly assignments “quests” is fine. However, I feel that calling actual concepts by these names is more confusing than anything. For example, calling bind groups “binding spells” just makes things more confusing.’

4.2 Mid-Semester Survey 2: 3D Graphics Phase

The second survey (11 of 17 students; 65% response rate) was distributed after the second trial, following the completion of the 3D graphics scrolls. Student sentiment remained generally positive but showed a decrease in motivation as the course content became increasingly complex mathematically. Only 73% of students rated their enjoyment of the narrative format at a higher level (4 or 5), and two students now indicated a preference for traditional formats. Just 64% found the story elements helpful in understanding 3D topics (ratings 3 or above).

Nonetheless, 91% of students still found the quest format helpful for structuring their learning (ratings 3 or above), and several highlighted its role in supporting shader fluency and experimentation. However, open-ended responses revealed a recurring need for more direct technical instruction. Students noted difficulty connecting fantasy framing with shader logic because of some “confusing terminology”. Despite these challenges, many continued to appreciate the freedom to approach quests creatively and incrementally. One student

noted: ‘Yes, I think narrative-driven learning could be beneficial in other CS classes to make courses more engaging and memorable, and students can refer back to previously learned topics more easily if they are more memorable in a narrative-driven format.’

4.3 Final Course Evaluation

The final university-administered evaluation, conducted with 12 of 17 students (70.6% response rate), provided a comprehensive view of student learning outcomes and perceptions. All respondents reported learning “a lot” or “a great deal”, and 75% indicated moderate or strong proficiency in WGSL shader programming. Confidence levels remained high for 2D graphic concepts (75%), though dropped for 3D content (50%).

Course design elements also received strong endorsement. 83% of students found the scrolls helpful or very helpful for progressive learning, and 92% agreed that the quests supported shader development. 83% reported that the final enchantment provided an effective opportunity to synthesize skills and learn from peers. Open-ended feedback highlighted a sense of accomplishment alongside concerns about workload and pacing. One student reflected, ‘I was greatly challenged in this course and faced roadblocks I never thought I could get through.’ These responses align with mid-semester surveys and reinforce the importance of striking a balance between narrative framing and structured scaffolding.

5 Lessons Learned

The evaluation of CGWA through two mid-semester surveys and a final course evaluation outlined several important takeaways for the next iteration. While the responses demonstrated strong engagement and significant learning gains in the course’s narrative and scaffolded structure, the feedback also highlighted key challenges related to pacing, technical clarity, and workload distribution.

5.1 Strengths

Students overwhelmingly reported that they learned “a great deal” from the course and rated the instruction as very effective or effective. The scrolls and quests were frequently cited as helpful in building both confidence and proficiency in shader programming, with over 90% of students indicating that weekly quests were effective in helping them become proficient in WGSL.

The narrative framing and quest-based structure were well received, with students noting that the course felt cohesive and thoughtfully designed, particularly in how the scrolls, quests, and final enchantment built on one another throughout the semester. Multiple students noted that they were pushed to

overcome challenges they had ‘never thought [they] could get through’, and that they were proud of the work they produced, especially in the final enchantment.

5.2 Challenges

A recurring theme in student feedback was the high cognitive load and fast pace of the course. Several students expressed that the workload, particularly in later 3D content, was intense and suggested that the course might be split into two parts (2D and 3D) to allow for deeper engagement. Others recommended simplifying or reducing the number of quests or providing more targeted technical instruction on WGSL (shader programming) to reduce ramp-up time.

Scrolls were effective as learning tools, but some students found them challenging to interpret without additional guidance or examples. Suggestions included adding more example code, embedding inline documentation, and supplementing with optional video walkthroughs. The final enchantment was widely valued, but its time demands were also flagged. One student estimated it required over 40 hours of work and suggested spreading it over more weeks or integrating debugging time more explicitly.

6 Conclusion and Future Directions

CGWA demonstrated that a narrative-driven, shader-first approach can make computer graphics more accessible, engaging, and creatively rewarding, particularly for students without prior exposure to advanced mathematics. The scroll–quest–trial structure sustained motivation, supported conceptual understanding, and encouraged students to take ownership of their learning through open-ended exploration and visual experimentation. While the narrative framing was most effective during the 2D phase, the underlying structure continued to support learning even as the content became more complex.

Looking ahead, future offerings will focus on enhancing technical scaffolding, adjusting pacing, and expanding asynchronous support. Scrolls will be revised to include more guided shader mini-lessons and example-driven explanations, especially for WGSL and projective geometric algebra. The 3D phase will be modularized to ease the conceptual load and allow for more focused progression. To better support self-paced learning, additional resources such as annotated code and concise instructional videos will be developed. The final enchantment timeline will also be restructured to include phases for planning, debugging, and peer feedback. Together, these changes aim to preserve the narrative-driven approach and creative foundations while strengthening the scaffolding needed for all students to succeed.

Acknowledgment

I would like to express my sincere gratitude to the students - *Aiden Ren, Clea Ramos, Jackson Rubiano, Jesse Tomaskovic, Kevin Duong, Luke Snyder, Nolan Sauers, Ramon Asuncion Batista, TJ Freeman, Titus Weng, and others*, who generously gave their consent for the use of screenshots of their quests and projects in this paper for demonstration purposes. Your creativity and openness to share your work have enriched this article, and I am proud to highlight your contributions.

References

- [1] H. Fink, T. Weber, and M. Wimmer. “Special Section on Eurographics Education: Teaching a modern graphics pipeline using a shader-based software renderer”. In: *Comput. Graph.* 37.1–2 (Feb. 2013), pp. 12–20. ISSN: 0097-8493. DOI: 10.1016/j.cag.2012.10.005. URL: <https://doi.org/10.1016/j.cag.2012.10.005>.
- [2] Caitlin Kelleher, Randy Pausch, and Sara Kiesler. “Storytelling alice motivates middle school girls to learn computer programming”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’07. San Jose, California, USA: Association for Computing Machinery, 2007, pp. 1455–1464. ISBN: 9781595935939. DOI: 10.1145/1240624.1240844. URL: <https://doi.org/10.1145/1240624.1240844>.
- [3] Dejan Nikolic and Ching-Kuang Shene. “GraphicsMentor: a tool for learning graphics fundamentals”. In: *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education*. SIGCSE ’02. Cincinnati, Kentucky: Association for Computing Machinery, 2002, pp. 242–246. ISBN: 1581134738. DOI: 10.1145/563340.563432. URL: <https://doi.org/10.1145/563340.563432>.
- [4] Jennifer Parham-Mocello et al. “Story Programming: Explaining Computer Science Before Coding”. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. SIGCSE ’19. Minneapolis, MN, USA: Association for Computing Machinery, 2019, pp. 379–385. ISBN: 9781450358903. DOI: 10.1145/3287324.3287397. URL: <https://doi.org/10.1145/3287324.3287397>.
- [5] Michael James Scott and Gheorghita Ghinea. “Integrating fantasy role-play into the programming lab: exploring the ‘projective identity’ hypothesis”. In: *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. SIGCSE ’13. Denver, Colorado, USA: Association for Computing Machinery, 2013, pp. 119–122. ISBN: 9781450318686. DOI: 10.

- 1145/2445196.2445237. URL: <https://doi.org/10.1145/2445196.2445237>.
- [6] Lee Sheldon. *The Multiplayer Classroom: Designing Coursework as a Game*. 1st. Boston, MA, USA: Course Technology Press, 2011. ISBN: 1435458443. DOI: 10.5555/2031502. URL: <https://dl.acm.org/doi/10.5555/2031502>.
- [7] Jerry O. Talton and Darren Fitzpatrick. “Teaching graphics with the OpenGL shading language”. In: *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*. SIGCSE ’07. Covington, Kentucky, USA: Association for Computing Machinery, 2007, pp. 259–263. ISBN: 1595933611. DOI: 10.1145/1227310.1227402. URL: <https://doi.org/10.1145/1227310.1227402>.