Technical Scaffolding for Interdisciplinary Collaboration in Serious Games

Ali Rezaei

February 4, 2026

1 Project role and context

This case study covers my technical consulting work on "Mudbricks and Pixels," an ongoing interdisciplinary project (September 2025 to present) that simulates Adobe and mudbrick construction processes. The project aims to democratize architectural knowledge by building a VR and web-based program that teaches fundamental construction skills to women in different parts of the world. The core team included two undergraduate design students, a supervising professor, and me as the lead technical consultant.

My main responsibility was to create a development environment that let non-technical stakeholders participate directly in the Unity workflow. Rather than only implementing requests, I acted as a technical translator and built infrastructure that let designers bring assets, from custom models to 3D scans of real mudbrick sites—into a physics-driven environment without requiring them to be fluent in C#.

2 Collaboration challenges: the vocabulary of interaction

Early collaboration was slowed by a vocabulary mismatch between engineering specs and design goals. Designers focused on narrative and tactile realism: how a mudbrick should feel heavy, or how a surface should respond to tools. Unity's physics engine, by contrast, relies on mass, drag, and friction.

To the designers, the physics system felt chaotic rather than creative, and initial attempts to communicate with raw documentation created a barrier to entry. To support the instructional goals of the project, we had to demystify "physics simulation" and reframe it as a set of predictable, adjustable creative controls.

3 Methods: documentation as infrastructure

I addressed the gap using pedagogical documentation and analogical scaffolding. We built internal guides that explained abstract behavior with metaphors the design students already understood.

3.1 Metaphorical translation

In the internal handbook, *Working with Physics in Unity*, I avoided vector math and leaned on physical analogies. For example, "drag" became an "air brake" for objects rather than an air-resistance coefficient. A "kinematic" rigidbody was described as a "ghost driver" that stays on rails and can push other objects but is not driven by forces. Colliders were framed as “protective bubbles” that define physical shape.

3.2 Inspector-first workflow

I promoted a workflow where the Unity Inspector was the primary tool for designers. The documentation emphasized that physics can be “set and forget”: gravity and collisions are handled automatically once values are configured. This let designers use Physics Materials to tune “bounciness” and “friction” to distinguish surfaces (for example, “ice” versus “sandpaper”) without writing code.

3.3 Scaffolded interaction models

We also distinguished between “teleporting” objects (direct transform edits) and “pushing” them with forces. The distinction mattered most for VR, where natural motion is key to immersion.

4 Results

Design students were able to populate scenes with interactive assets and configure Rigidbodies and Colliders to match the physical properties of mud and stone. Communication improved as the team adopted the shared metaphorical language. Conversations shifted from vague complaints about “glitchy” motion to precise requests about collision layers and trigger zones.

5 Lessons learned and recommendations

Based on the “Mudbricks and Pixels” collaboration, these practices are recommended for engineers working with creative teams in digital heritage contexts:

- Establish a “pidgin” language early by mapping technical terms to domain-specific metaphors (for example, triggers as sensor regions or tripwires).
- Treat documentation as a product. Internal guides should be delivered alongside code and use a friendly, second-person tone to reduce anxiety.
- Scaffold the “happy path” with tools and presets so designers can focus on creative variables. Explain common misconceptions, such as mass affecting collisions but not fall speed.
- Prioritize one-click feedback. Designers need immediate visual confirmation, so encourage frequent use of Play Mode.
- Layer complexity over time. Start with basic collisions (solid objects) before introducing triggers (sensors) or collision layers.
- Contextualize constraints. Frame settings like Fixed Timestep or Solver Iterations as trade-offs between smoothness and accuracy, not math problems.

6 Limitations

This reflection is based on qualitative observation, and the project is still ongoing. We did not run formal pre- and post-tests on technical literacy. The collaboration also depended on the specific constraints of "Mudbricks and Pixels," which prioritized physics-based spatial interaction over complex algorithmic logic. Projects that require heavy custom scripting may encounter different hurdles when bridging the engineer-designer divide.