

Metaball Madness - The Rigging Of An Implicit Surface Character

Anna-Christine Lykkegaard
Pixar Animation Studios
Emeryville, USA
acl@pixar.com

Andrew Butts
Pixar Animation Studios
Emeryville, USA
andru@pixar.com

Julian Teo
Pixar Animation Studios
Emeryville, USA
jteo@pixar.com

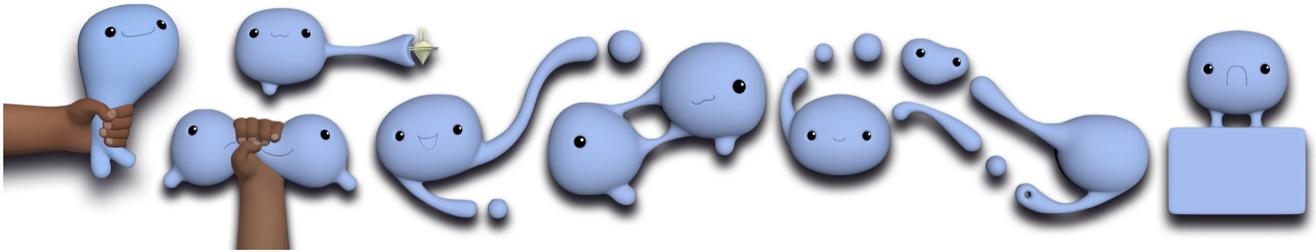


Figure 1: Showcasing early implicit exploration of capabilities. ©Pixar

Abstract

We present a novel character rigging solution developed for OOOOO, a liquid supercomputer in Pixar’s Elio.

OOOOO’s design and desired movement necessitated reimagining our conventional way of articulating characters and she became Pixar’s first mesh-free character rig. We developed a system that allowed our animators full fidelity control over what is essentially a rigged shader while ensuring downstream renderability. [Luo et al. 2025] The system’s architecture supports a hierarchical arrangement of implicit surface primitives and operators, allowing for complex transformations while preserving normal animation paradigms and offers unprecedented flexibility in character animation.

ACM Reference Format:

Anna-Christine Lykkegaard, Andrew Butts, and Julian Teo. 2025. Metaball Madness - The Rigging Of An Implicit Surface Character. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH Talks '25)*, August 10–14, 2025, Vancouver, BC, Canada. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3721239.3734079>

1 Background

From the early concept art and story explorations, we knew this character required articulation beyond the capabilities of our traditional rigging system: sliding arms anywhere on her body, separating chunks of liquid from her main form, absorbing objects, adding arms on the fly, connecting her arms to screens or turning into a screen, and the ability for her facial features to slide around her body.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGGRAPH Talks '25, Vancouver, BC, Canada
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1541-9/25/08
<https://doi.org/10.1145/3721239.3734079>

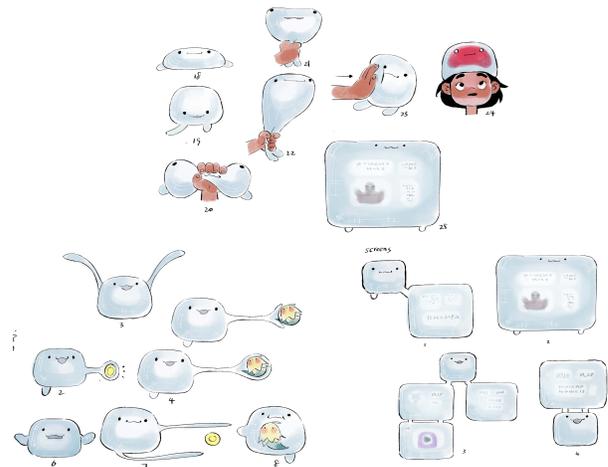


Figure 2: Showcasing some early concept art of OOOOO by Yingzong Xin. ©Pixar

2 Development

Our existing software lacked the ability to build and deploy implicit surfaces at scale, necessitating new primitive types for raymarched implicits and their visualization.

We introduced two types of prims: BIObjects - GLSL code for SDF computation to create simple primitives (e.g., ellipsoids, rounded boxes, tori) with the option for custom SDFs (e.g., mouth, eyelids, arms), and BOperators - which apply smoothing, subtracting, or intersecting operations to child BIObjects or the ability to apply other types of operations like sliding on SDF and squetching.

These are structured under a new material type, BIPreviewMaterial, which, along with BIPreviewCanvas, forms the basis for creating and deploying the raymarched implicits and feeding the computed shader hierarchy.

Turning off viewport materials will hence reveal the only polygonal component of OOOOO - the raymarched low-poly bounding sphere in which OOOO's shape is calculated within. For optimization purposes given the screen based approach, we added an adjustable computation limit after the first hit, which could be upped in complex shots if needed.

3 Modeling

Our characters are historically modelled with traditional topology and either modelled or imported into a maya scene to be consumed downstream. Yet the only thing existing in OOOOO maya file is a number of crude sbdv's used solely for the ease of picking parts of OOOOO during animation to uphold the expected control scheme using existing toolsets requiring geometry.

OOOOO is the first character modeled inside Presto. Unlike traditional modeling, where modeling precedes rigging, we defined OOOOO's control scheme and features first and then arranged the implicits to achieve her desired design.

The hierarchical method of building allowed us to add complexity dynamically in shots, offering a uniquely flexible character.

4 Rigging

After the initial development of the hierarchical implicit system, the rigging to drive the implicit shapes was kept simple. BObjects were

hooked up to rigged pivots, and BOperator values were connected to attributes, forming a surprisingly simple rigging base.

Her eyelids and mouth used a control scheme similar to our standard setup but built upon spline rigs to enable uniquely requested shapes—such as geometric mouth and lid shapes. The GLSL behind these features are extracted based on curves and then smooth intersected to form cavities for the eyes and mouthbag.

Her arms are rigged curves, with the GLSL using smoothly unioned SDF spheres to form blobby "noodle" limbs with dynamic thickness control based on the arm-curves. OOOOO was designed for maximum flexibility in mind; having the ability to toggle on/off shapes, limbs, and eyes on the fly, treating her like a stop-motion clay puppet - adding or subtracting elements as needed.

We also developed native SDF meshing in Presto, though its performance cost kept it out of regular animation use.

Unexpectedly, OOOOO's tech found broader use. One animator repurposed her black silhouette to explore the design of the Alien Ambassador, Auva. It was also used in the TranslatorDisk, CloningClay mockups, and to prototype ideas in other parts of the film.

References

- Cat Luo, Trent Crow, Fernando De Goes, and Ferdi Scheepers. 2025. Metaball Madness: Look Development For A Shapeshifting, Implicit Surface Character On Pixar's Elio. In *ACM SIGGRAPH 2025 Talks*. ACM, 2 pages. doi:10.1145/3721239.3734130