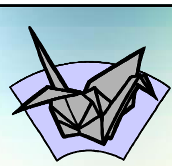


Ori

An Interactive Paper Folding Simulation

Purdue University
CS 490T
Tim Thirion, Darin Rajan, and Jennifer Lin

Overview

- Motivation: The History of Origami
- Previous Research
- The Objective
- Data Structures and Operations
- Implementation Issues
- Demo
- Conclusion

History

- Originated in Japan several hundred years ago
- The practice of paper folding was divided into:
  - **Recreational**
  - **Ceremonial**

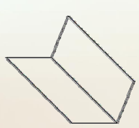
- The name origami was coined in 1880 from the words oru (to fold) and kami (paper). Previously, the art was called orikata ("folded shapes").
- Today, hobbyists and professional paper folders are spread throughout the world

Previous Research

- Mathematical Research
  - Huzita's Axioms: Provide a generic mathematical framework that describe all origami folds.
  - Erik Demaine (MIT) – Developed several theorems related to paper folding and cutting.
  - Protein folding in biochemistry.
- Interactive Origami Simulations
  - Very few (is this a warning?)
  - Only a 2D application could be found

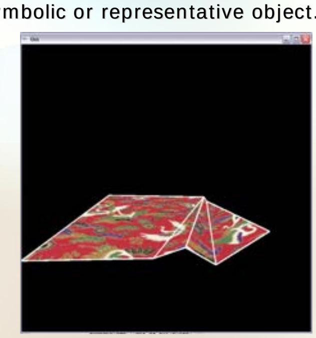
Objective

- To create a system that allows the user to create simple, 3D origami objects using an intuitive interface.
- The goal is to design a system such that basic folds are immediately available and complex folds can be added.



Goal

- Starting with a square of virtual paper, the user interactively folds until forming a symbolic or representative object.



Data Structures and Operations

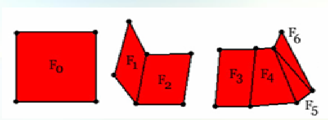
- How do we represent an origami computationally?
  - Many different ways to do it
  - Essentially an origami is a set of polygons
  - Trees and lists are two ways of storing the polygons.
  - This is a tough decision…

Connectivity

- Observation: An origami is highly connected.
- It has three data components:
  - Vertex – A point in space
  - Edge – A connection between two vertices
  - Face – A sequence of edges that form a closed polygon
- An origami is then approximated by a set of faces.

Conceptual Examples

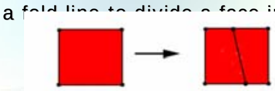
- A sequence of simple bends:



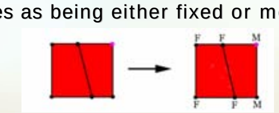
- Also, vertices keep track of the edges that are incident upon it.
- Edges keep track of the faces to the left and to the right.
-
- A “Winged Edge Boundary Representation”

The Three Core Operations

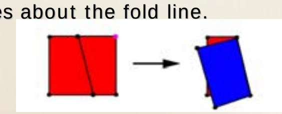
- Make a fold line to divide a face into two.



- Select a vertex and use it to label all of the vertices as being either fixed or moving.



- Rotate interactively all of the “moving” vertices about the fold line.



Complexity Issues

- It is difficult to determine if the sequence produced a fold acceptable to the rules of origami:
  - It cannot self-intersect.
  - It cannot be cut.
  - Many other constraints…
- A few degenerate sequences can be determined from the input sequence and state of the data structures.

Complexity Issues

- Implementation of visual effects
  - Advanced lighting, advanced shading, and shadowing were completely dependent on the origami data structure.
  - As the data structure changed, the algorithm for these effects had to change
- Many are tough to find and remedy.
- Because of this, the simulation is often easily broken.
- Undo and Redo help tremendously when this occurs.

Demo

Conclusion

- Research in origami is geared toward producing mathematical theorems, not interactive simulations.
- For our system, simple folds are easy; sequences of simple folds are difficult.
- Trying to catch degenerate cases is difficult.
- Debugging involves tracing through the *entire* structure and discerning the unique circumstances that caused the crash.

Questions, Comments, Rotten Fruit?

- tthirion@purdue.edu
- rajand@purdue.edu
- linjs@purdue.edu