Origami project overview and plan

Contents

[Overview 2](#_Toc31836179)

[Version 1: 2](#_Toc31836180)

[Paper: 2](#_Toc31836181)

[Nodes: 2](#_Toc31836182)

[Geometric rules: 3](#_Toc31836183)

[1: all space must be filled. 3](#_Toc31836184)

[2: no area is cut off. 3](#_Toc31836185)

[Generating creases 3](#_Toc31836186)

[Diagonal creases: 3](#_Toc31836187)

# Overview

This java applet can generate a crease pattern, which collapses into a given base. The applet can take a design from the user, or load a saved file. The program is made in three versions, each building on the last.

1: basic layout, place nodes, save and display paper.

2: check geometric rules, and generate creases

3: optimize placement of nodes and cuts, then generate creases.

# Version 1:

## Paper:

The paper class stores the data for a given design. This consists of all the nodes, creases, and cuts, as well as a name/UID. Each paper is divided into a grid of evenly sized squares, which are used to center and plan the nodes. This grid will be divided into half for pre-creasing, and diagonal creases will be added.

## Nodes:

A node is the building block of a origami design. Each has a size, N, which is the amount of space that it will take up on the finished base. Each one has a UID, but no data about other nodes or creases. Nodes come in three types: leaf nodes, thin nodes, and rivers.

Leaf nodes are the most common. one the center is placed (by hand or by optimization), it fills the surrounding area. This area is filled in a square at least N squares in all directions, ending on an edge or cut. This area cannot be used for any other nodes, and it cannot be cut.

Thin nodes are rare, but useful for antennae and long thin details. They take up an area of a length N, and a width of 2 squares.

River nodes separate sections of the base away from each other. They start at the symmetry line, and extend until they run off an edge or cut. They have a size N, and must maintain a constant width throughout. They can take 90 degree turns, but must be constant width.

# Geometric rules:

1: all space must be filled. If there is unfilled space, the initial version should warn the user. The final version might suggest solution. In either case, the space must be filled by one of five methods. Nearby leaf nodes or thin nodes could be expanded to incorporate the space, which often makes then non-square. An adjacent river can wander into and out of the space, like a cul-de-sac. In these cases, the empty paper is unused in the final design, and adds to loss and inefficiency. The final solution is to add an extra node into the unfilled space, which may be useful for detailing or patterning in the final design. This is often not possible/practical, if the space is small or oddly shaped.

2: no area is cut off. Though this seams obvious, this is very important when placing cuts, since the final design is one base. Though it might be useful to cut off excess paper in some areas, it is much better to include all the paper into the design.

# Generating creases

## Diagonal creases:

Leaf nodes and thin nodes have similar structure for diagonal folds. Once the borders are defined, the angle bisectors are extended inwards from all corners. Note: for nodes near edges or cuts, the edges do not count as borders. When two 45-degree creases meet, they merge into a new horizontal crease. If two creases meet head on, they both end. Once all diagonal creases are done, the grid creases are made

Grid creases:

This step may be turned off, since it is trivial for experienced folder. In general, if the paper is dived into a M by M grid, this grid will be folded in valley folds. In between each of these folds is a mountain fold, dividing the grin into a 2\*M square grid of alternating creases. Some of these folds may not be necessary for the final design, but for the first version this is sufficient.