Technology
Arts Sciences
TH Köln

Origami Diagramming

Development of a desktop application for creating origami diagrams

Practical Project

by

Julian Hardtung

submitted to obtain the degree of

Bachelor of Science (B. Sc.)

at

TH Köln University of Applied Sciences
Campus Gummersbach
Institute of Informatics and Engineering

Course of Studies

Media Informatics

First supervisor: Prof. Dr. Martin Eisemann
TH Köln University of Applied Sciences

Second supervisor: Matthias Groß
TH Köln University of Applied Sciences

Gummersbach, February 10, 2020

Adresses: Julian Hardtung
Lachtstraße 12
51645 Gummersbach
ju.hardtung@gmx.de

Prof. Dr. Martin Eisemann
TH Köln University of Applied Sciences
Institute of Informatics and Engineering
Steinmüllerallee 1
51643 Gummersbach
martin.eisemann@th-koeln.de

Matthias Groß
TH Köln University of Applied Sciences
Institute of Informatics and Engineering
Steinmüllerallee 1
51643 Gummersbach
matthias.gross2@th-koeln.de

Contents

1 Introduction

Creating instructions for folding Origami models is a tedious and time consuming task. These so called origami diagrams (see Fig **??**) have to be accurate representations of the paper for every folding step, in order to unambiguously explain how to fold the model. Every flap, crease and edge has to be drawn (see Section 2.1 for exceptions), which makes the process slow and especially error-prone for complex models. So far, diagrams got either drawn by hand, or created with the help of digital vector programs like Inkscape [1] or Adobe Illustrator [2]. Even though the digital diagramming helped with the accuracy of the finished instructions, the task itself was still quite time consuming.[**?**]

Up until the point of this publication there is no publicly available program that offers specific features for the origami diagramming process. The previously mentioned vector programs can be used for diagramming, however as they weren't developed with origami diagramming in mind, there are a lot of shortcomings in their feature set. These shortcomings of current programs will be further elaborated on in Section 3.

The goal for this project is to develop a desktop application that implements features specifically for the origami diagramming process. The standardized symbols and overall notations have to be included and the program has to offer functions that increase the efficiency of creating diagrams.

This work starts with an overview and subsequently a categorization of current origami diagramming notations. Based on these findings, a requirements analysis can be carried out in order to define the required features of the planned program. Any problems or other findings during the development process will be documented so that they may be beneficial to others in the future.

[1] https://inkscape.org/
[2] https://www.adobe.com/de/products/illustrator.html

2 Diagramming Notation

In order to define the concrete requirements of the planned diagramming program, all commonly used diagramming symbols and conventions have to be collected and categorized. After that groundwork a plan can be established on how to implement these findings in a desktop application.

The very fundamentals of Origami diagramming were developed and proposed by Akira Yoshizawa in his book *Atarashi Origami Geijutsu (New Origami Art)*[2] in 1954, which introduced a system of folding notation. Yoshizawas diagramming system is still widely used today and most commonly known as the *Yoshizawa–Randlett system*.

Despite its high popularity, different nuances and slight changes were made by different origami diagrammers. To avoid further confusion, especially for beginners, american physicist and origami artist Robert J. Lang compiled notations that were in use by different folders and proposed a standard for all different folding sequences. In order to support his claims, Lang sent "a questionnaire to 25 diagrammers around the world"[1] and tried to coherently argue in favour of, or against the various results.

Langs efforts were made under the assumption, that "[...] unless there is pressing reason otherwise, we should use the standard notation developed by Yoshizawa" [1].

To start off we have to define what exact components a diagram consists of. Most importantly, there always is a visual representation of the paper in the current step. This representation should show all or most (see Section 2.1) on when to leave out things creases, flaps, edges and layers to accurately show how the actual paper model would look like. Secondly, there has to be a description of the actual folding sequence for one step. This description is comprised of a textual explanation and a visual display of the step with the diagramming symbols (Section 2.2). Both the verbal and the visual instructions should be able to stand for themselves, although that might not always be possible for complex steps. More detailed rules and specific terms for the verbal instructions are described on Section 2.1.

2.1 General diagramming rules

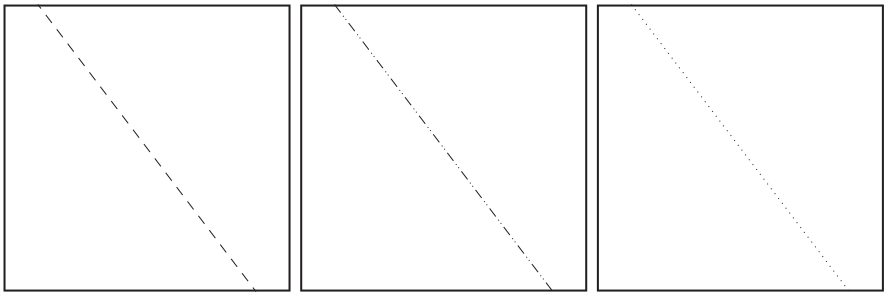
- Be consistent (stick with one notation, e.g. -..-..-.. or -.-.-.-. for mountain folds)
- edges = 1 point line; creases 1/2 point line; creases should not contact the edges that the creases end upon; DO touch edges that they go under
- use right origami grammar
- show one white and one colored side of the paper
- distort the model to show all the layers
- enlarging the diagram with a circle or just by common sense with good diagrams
- right numbering of steps 1,2,3,4 and 1a,1b,1c to explain hard steps (ony to elaborate on a single step)

2.2 Diagramming Notation

As there are quite a few simple and self-explanatory symbols, there will only be a further explanation if there are any additional rules or things to keep in mind while using them. Furthermore, on occasion Lang gives multiple valid symbols or ways to visualize a certain fold or folding sequence. The goal in these cases should be to give the user a choice on what method to use for a diagram.

2.2.1 Folds

Starting with the very basics in origami, there has to be a differentiation between the types of folds. Generally, there are 2 different kinds, namely the Valley Fold and Mountain Fold, which are shown in Figure 2.1a and Figure 2.1b respectively. Additionally the so called X-Ray Line is shown in Figure 2.1c. Even though it is not a real fold, it was added here to show the three different forms a line can be drawn in origami diagrams. In order to indicate what kind of fold the X-Ray line is representing, you should extend the line past the paper and use a valley fold or mountain fold line.



(a) Valley Fold (b) Mountain Fold (c) X-Ray Line

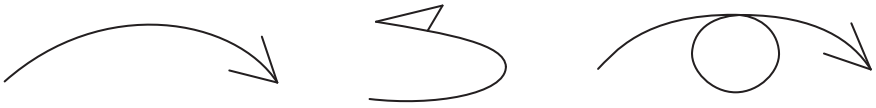
Figure 2.1: Different Lines in Origami Diagramming

Important for the displaying of lines is, that they don't start or end with a gap. This method ensures that there isn't any ambiguity on what reference points are needed for the fold.

2.2.2 Arrows

In order to show the folding steps unambiguously, there have to be arrows that indicate the direction the paper has to be folded. Showing only a valley or mountain fold leaves room for interpretation, which gets eliminated by

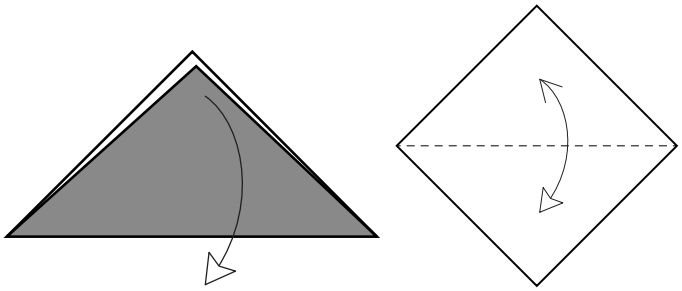
the addition of these specific arrows. In origami diagramming there are two groups, the Arrows of Motion and Arrows of Action. While the arrows of motion describe where the paper is folded to, the arrows of action indicate an action perfomed on the paper itself.



(a) Valley Fold (b) Mountain Fold (c) Turn the paper over

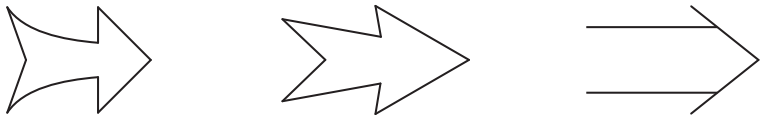
Figure 2.2: Arrows of Motion

The unfolding arrow can either stand alone and signalize an unfolding action, or be combined with a valley/mountain fold and indicate a folding and unfolding procedure.



(a) Unfold (b) Fold and unfold

Figure 2.3: Unfolding arrows



(a) Push here (b) Pull the paper out (c) Inflate here

Figure 2.4: Arrows of Action

2.2.3 Clarifying diagrams

Leader

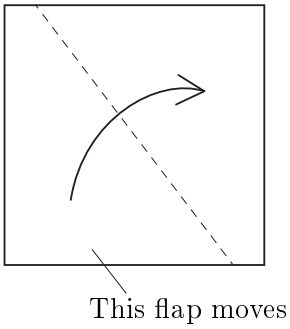


Figure 2.5: Use a leader to give additional information if needed

Equal distances

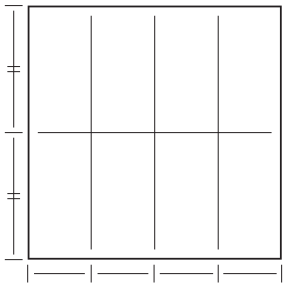


Figure 2.6: Equal Distances

Equal angles

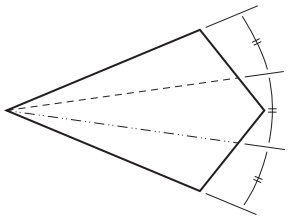


Figure 2.7: Equal Angles

Rotations

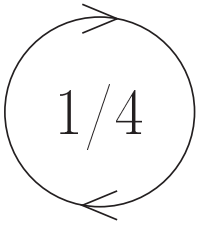


Figure 2.8: Rotation Symbol

X-Ray View

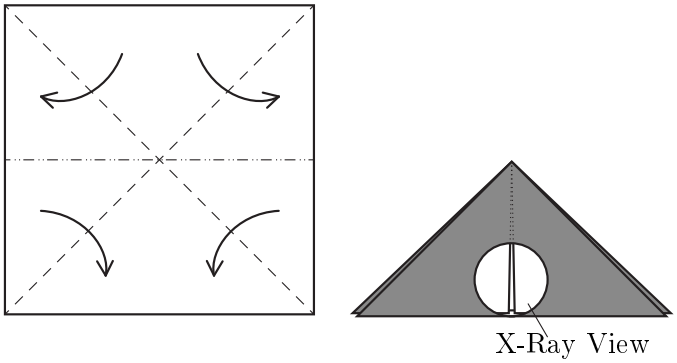


Figure 2.9: The X-Ray View shows hidden layers

Repetitions

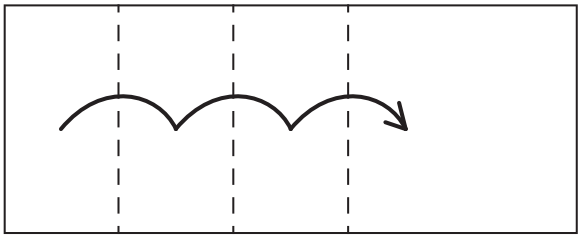


Figure 2.10: Fold over and over

For complicated repetitions you should show the result of the first fold that is to be repeated.
The repetition box from Figure 2.11 shows what exact steps are to be repeated and how many times.

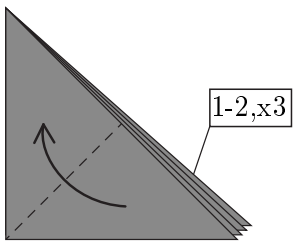


Figure 2.11: Repetition Box

Next view here

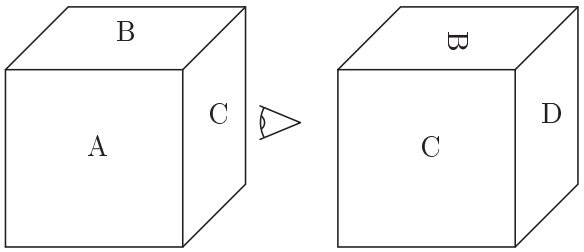


Figure 2.12: Next View Here

Hold here

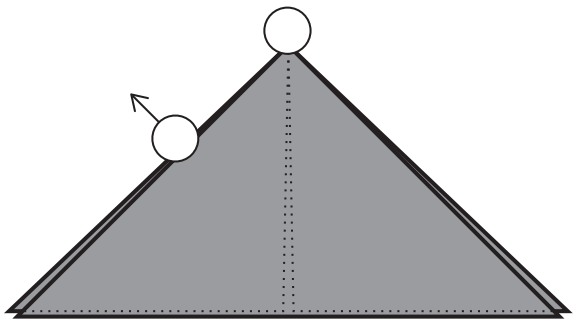
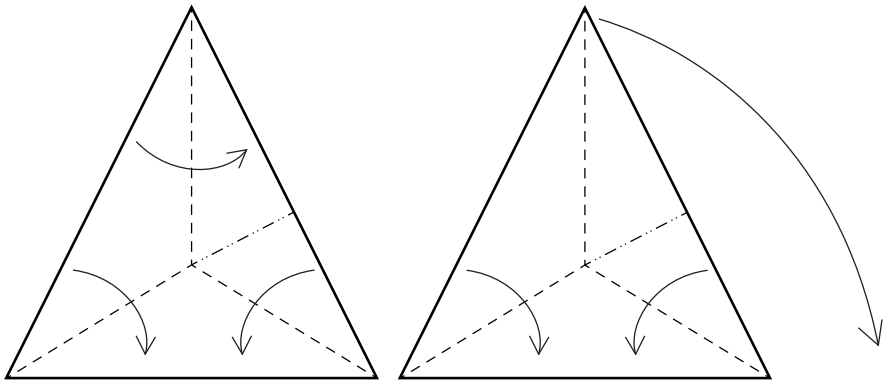


Figure 2.13: Hold here + hold here and pull

2.2.4 Folding Procedures

Rabbit Ear

Lang gives two valid methods to visualize a Rabbit Ear. While Figure 2.14a technically shows the more accurate format (according to the previously defined rules in Section 2.1), method B (see Figure 2.14b) is also desireable, as it shows the overall motion of the paper.



(a) Rabbit Ear Method A (b) Rabbit Ear Method B

Figure 2.14: Both methods show a rabbit ear fold

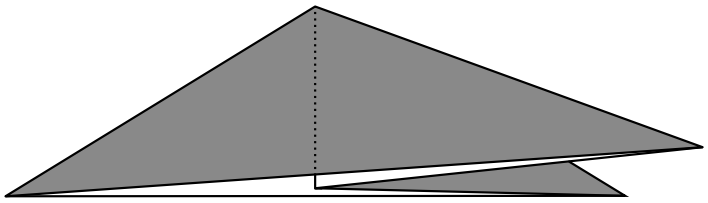
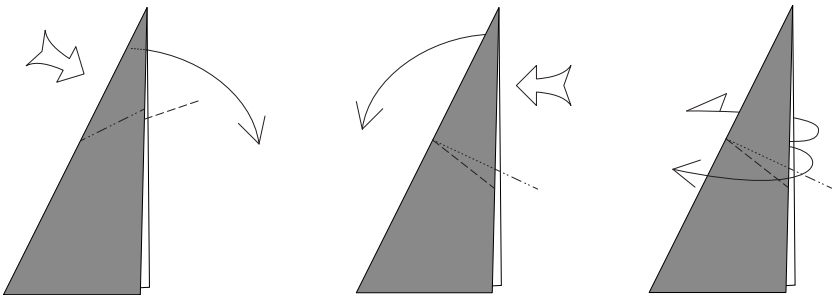


Figure 2.15: Rabbit Ear Result

Reverse Folds

Although Lang proposes two valid notations for outside reverse folds, for consistency's sake Method A should be used (see Figure 2.16b).

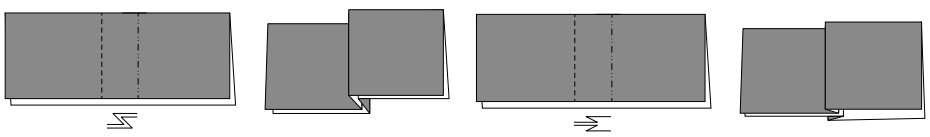


(a) Inside Reverse Fold (b) Outside Reverse Fold (c) Outside Reverse Fold

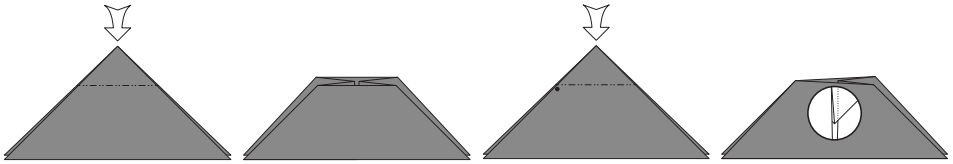
Figure 2.16: Different reverse folds

Crimping and Pleating

Sinks

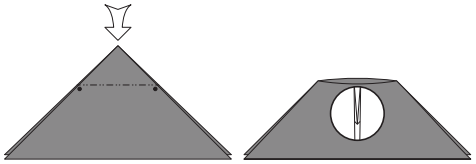


(a) Crimping (b) Pleating



(a) Open Sink

(b) Mixed Sink



(c) Closed Sink

3 Requirement Analysis

After collecting and categorizing all diagramming notations, the actual requirements of the planned system can be defined more precisely. On basis of the previous section (Section 2) three different requirement fields become apparent.

- Diagramming Symbols
- General Diagramming Rules
- General Program Tools

3.1 Diagramming Symbols

Beginning with the most important field, the diagramming symbols form the basis of diagramming and implementing all of them is imperative. Without all the different lines, arrows and other symbols there will be ambiguity in the later folding instructions. For a comprehensive list of all required diagramming symbols see Section 2.2.

3.2 General Diagramming Rules

This field contains more abstract rules and best practices while creating origami diagrams. Including them in the system has to be decided on a case to case basis, as some rules have to be kept in mind by the artist during diagramming. Things like using the right *origami grammar* or *distorting the model in order to show otherwise hidden layers and edges* (see Section 2.1) rely heavily on the experience of the artist.

Though in the future some of these rules could be displayed as a hint within the program. And in the case of the right origami grammar, it could be thinkable to develop a system that automatically adds the description text of a step based on the symbols used.

3.3 General Program Tools

Outside of providing all the diagramming symbols and including the general diagramming rules where possible, the program has to offer common tools that have become standard in most vector programs.

- **Select, Move, Resize, Rotate:** editing an object (line, symbol etc.)
- **Delete:** delete diagramming symbol or whole step
- **Duplicate:** duplicate diagramming symbols or whole step
- **Copy & Paste:** copy or past an object (line, symbol etc.)
- **Undo & Redo:** undo or redo the last action (adding a line, symbol etc.)
- **Zoom, Rotate, Flip:** editing a step
- **Save, Open:** saving and opening an origami diagram
- **Export:** exporting a finished diagram to e.g .pdf

Even though these features are quite important for the later usability of Origrammer, the focus was first set on implementing all origami specific tools first, as this provides the biggest unique feature set when comparing to already existing vector programs.

4 Program Features and Use

The Origrammer structure follows a similar approach to the real life diagramming process. To create a new model, we start with some general decisions. When selecting the "File > New" menu item, a new dialog box opens (see Figure 4.1), where you can choose *paper shape, size* and the *paper color* of both sides. Additionally, the text fields *Title, Author, Comments* can be filled out to provide further information of the model. These options can also be accessed and changed during the diagramming process (except for the paper shape) when selecting "Edit > Model Preferences".

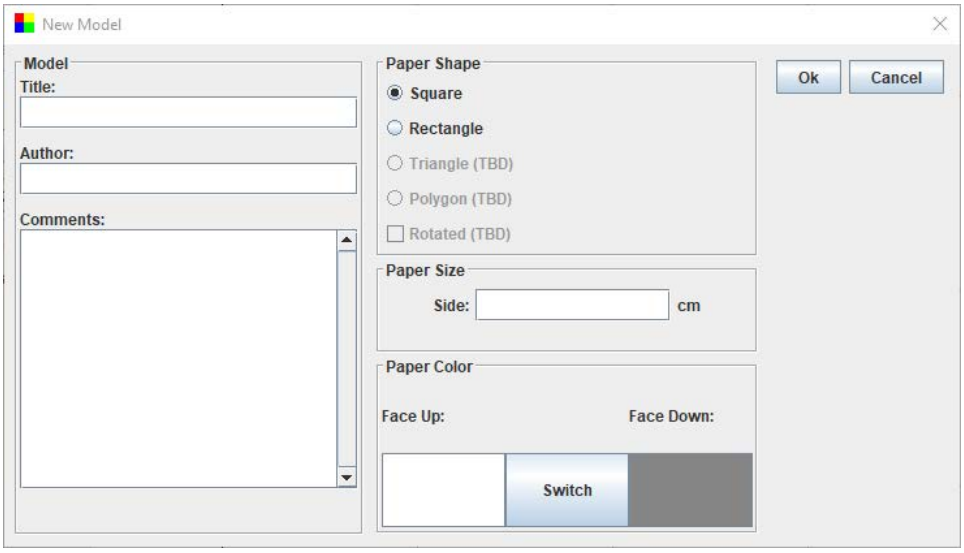


Figure 4.1: New Model dialog box

After specifying all the model settings, the main diagramming window is presented to the user (see Figure 4.3). This main editing area consists of four segments:

Side Panel

The side panel presents the toolbar where the user can switch between inputting lines, arrows, general origami symbols, the selection tool (further explained later), a measurement tool, as well as a filling tool. Furthermore, the displayed grid can be adjusted or turned off completely and faces that got filled with the filling tool can be turned invisible.

Top Panel

The top panel is directly connected to the side panel, as it shows the

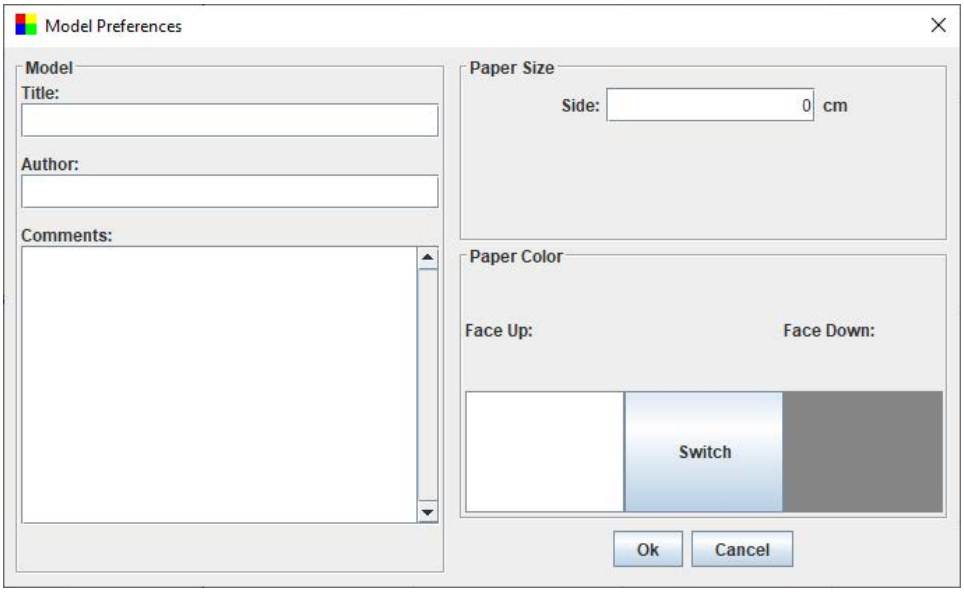


Figure 4.2: Model Preferences dialog box

actual settings, depending on what tool from the toolbar is active. Such as for the Line Input Tool, the top panel displays a JComboBox that contains the previously mentioned six different arrow types of origami. For other active tools this panel can show scaling or rotating options, different combo boxes, text fields or check boxes for a variety of adjustments.

Navigation Panel

Unlike the aforementioned panels, the navigation panel directly influences the model panel. As the name already states, this panel allows the navigation through the individual diagram steps. It also provides a JTextField to give the textual instructions for the current step.

Model Panel

The previously specified piece of paper is displayed in the model panel. This is the place where the paper, together with the arrows and other symbols is rendered to represent the current state of the paper for each folding step.

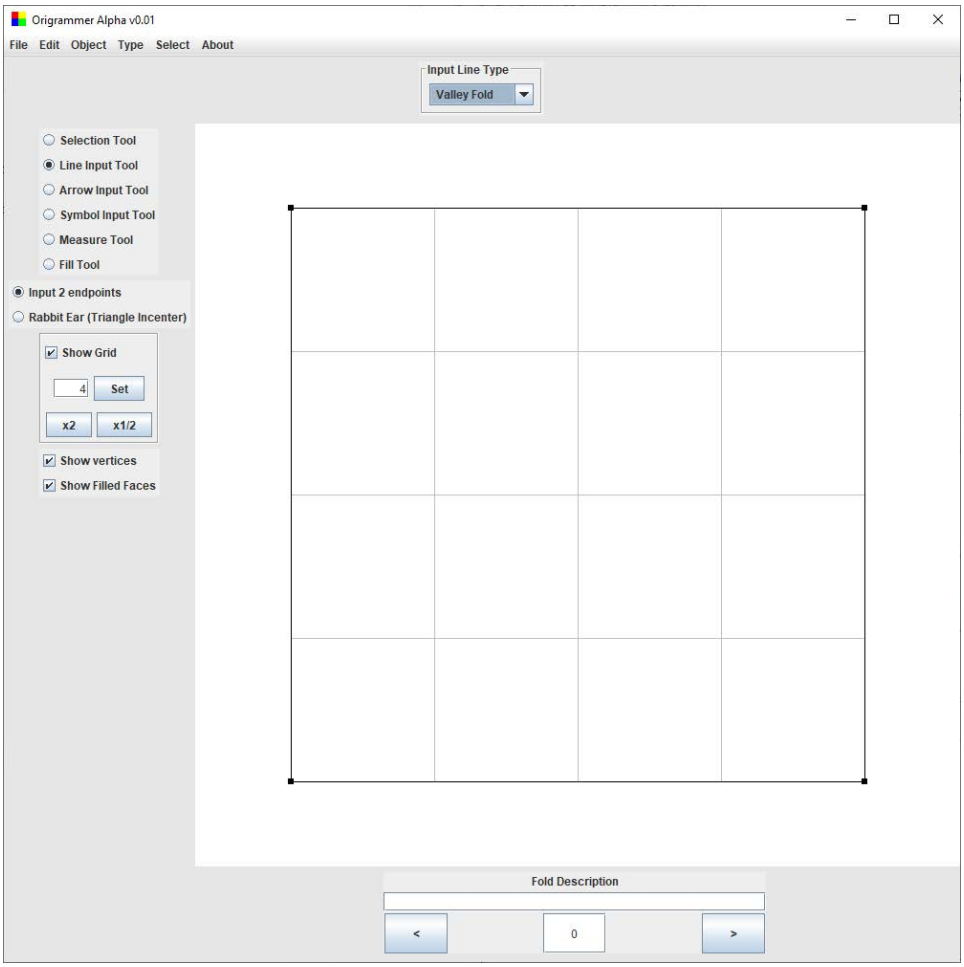


Figure 4.3: Origrammer main window

5 Problems & other findings

quality problems while scaling the notation symbols –> solution: using .svg instead of .png

rotating ImageIcons

JFrame –> JPanel –> JLabel –> ImageIcon –> BufferedImage –> Arrow

Can move, scale but not rotate (either override the paintComponent method) or use graphics2d.rotate();) Both doesn't work for my setup

dirty solution–> load different, already rotated images depending on Ori-Arrow rotation.

calculations are sometimes breaking depending on user input | –> for example OriEqualAnglSymbol: firstV –> point of angle scndV –> verticalVertex if verticalVertex and horizontalVertex are switched, everything breaks thirdV –> horizontalVertex

6 Prospect and further possible Features

Despite the mostly finished state of Origrammer there are various additional features that could prove useful once implemented. This section will list and explain missing features and general usability improvements and the following subsections are roughly sorted by descending importance. Though this should not be seen as an extensive list of missing essential features, but rather as a prospect on what the further development of this diagramming program could look like.

6.1 Export to .pdf

Exporting a finished diagram to the Portable Document Format (PDF) would provide an easy way to share diagrams created within the Origrammer. This would provide greater accessibility to the diagrams while also removing the requirement of using the Origrammer just for displaying purposes. This would be especially helpful for folders who simply want to follow the diagrams.

6.2 TBD

Glossary

Arrows of Action Arrows of Action indicate an action performed on the paper.. 8

Arrows of Motion Arrows of Motion indicate where the paper folded towards.. 8

Mountain Fold The fold that results after folding one part of the paper behind the other. After unfolding you roughly see an A-shape. 7

Origami (jpn: *ori* = *folding* and *kami* = *paper*) is the art of folding paper into models of animals, people and objects.. 4

Origrammer Origrammer is the name of the diagramming program, developed within the scope of this project. 21

Rabbit Ear A Rabbit Ear is a folding sequence, that narrows the paper and creates a new flap. This is done by creating three folds at the bisectors of a triangle.. 12

Valley Fold The fold that results after folding one part of the paper over the other. After unfolding you roughly see a V-shape. 7

X-Ray Line An X-Ray line indicates a fold or edge that is hidden behind a layer of paper. 7

List of Figures

List of Tables

References

[1] Robert J. Lang. Origami diagramming conventions, 2011. http://www.langorigami.com/article/origami-diagramming-conventions.

[2] Akira Yoshizawa. *Atarashi origami geijutsu.* Origami Geijutsu-Sha, Tokyo, 1954. no ISBN.