Technology
Arts Sciences
TH Köln

Origami Diagramming

Evaluating and Improving the Origami Diagramming Tool Origrammer

Bachelor Thesis

by

Julian Hardtung

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: Prof. Dr. Christian Kohls
TH Köln University of Applied Sciences

Gummersbach, August 4, 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

Prof. Dr. Christian Kohls
TH Köln University of Applied Sciences
Institute of Informatics and Engineering
Steinmüllerallee 1
51643 Gummersbach
christian.kohls@th-koeln.de

Contents

1 Abstract

2 Introduction

Origami is the Japanese art of folding paper into models of animals, people or other objects. Folding such models can be a complex task, so in order to make origami more accessible, a system of folding instructions was developed. The diagramming system developed by Akira Yoshizawa was first published in the book *Atarashi origami geijutsu* in 1954 [8]. Yoshizawa's origami notation introduced a set of symbols and arrows to explain the different folding actions. After further development by Sam Randlett and Robert Harbin this set of origami symbols and became the international standard[7] [**?**]. By creating these drawings for each folding step, even origami beginners could fold models. These origami diagrams (see Figure 2.1) 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, which makes the process slow and especially error-prone for complex models. While for example the crane model in Figure 2.1 can be folded in 17 steps, more complex models can take hundreds of steps to complete.

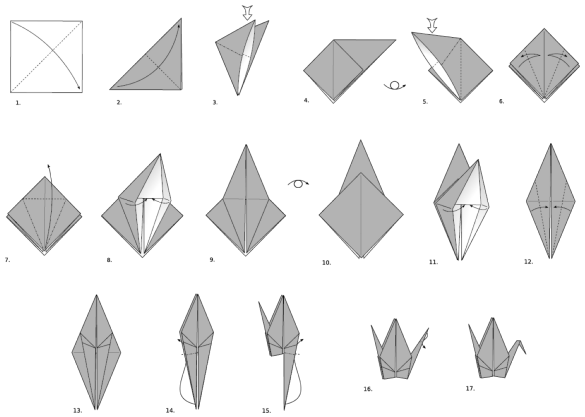


Figure 2.1: Crane Diagram - [Andrew Hudson 2011 [3]]

In order to help artists during the creation process for origami diagrams, the Origrammer [2] was developed.

The Origrammer is a desktop application, developed by the author, which offers specific features for creating such diagrams. This program simulates, how an origami artist would create diagrams by hand. Lines, arrows, and symbols can be placed and new diagram steps can be created. Addi-

tionally, helping features are included to speed up the diagramming process as a whole. A more detailed overview and explanation of the Origrammer can be found in Section 3.

This Bachelor Thesis now aims to evaluate the current state of this program and to develop it further to improve usability, efficiency and effectivity. This thesis will start with a short summary on how the Origrammer works and what a typical workflow looks like. Afterwards, an evaluation has to be carried out, in order to get a sense of the current state of this program. Flaws and missing features have to be explicitly defined, so that a plan can be created on how to further improve the Origrammer.

As the evaluation will potentially find a multitude of different types of flaws and problems, a specific focus has to be defined. In case of this thesis, the main goal is to maximise the speed of creating origami diagrams, while also offering a usable user interface with which also novizes can work with ease. This target was set with the original plan of the Origrammer in mind:

> "*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.***" [2]

Then after evaluating and collecting potential improvements, a more detailed plan can be created on how to actually implement these changes. Different possible solutions should be tested for pros and cons and the best solutions should then be developed for the Origrammer. Although not all potential improvements can be implemented within the scope of this thesis, all found flaws should be categorized and documented, which can help further development in the future.

3 Origrammer

This section will provide a more detailed overview over the Origrammer. It's parts will be explained and a typical workflow will be shown. Figure 5.1 shows the user interface of the Origrammer, which is split up into five major parts.

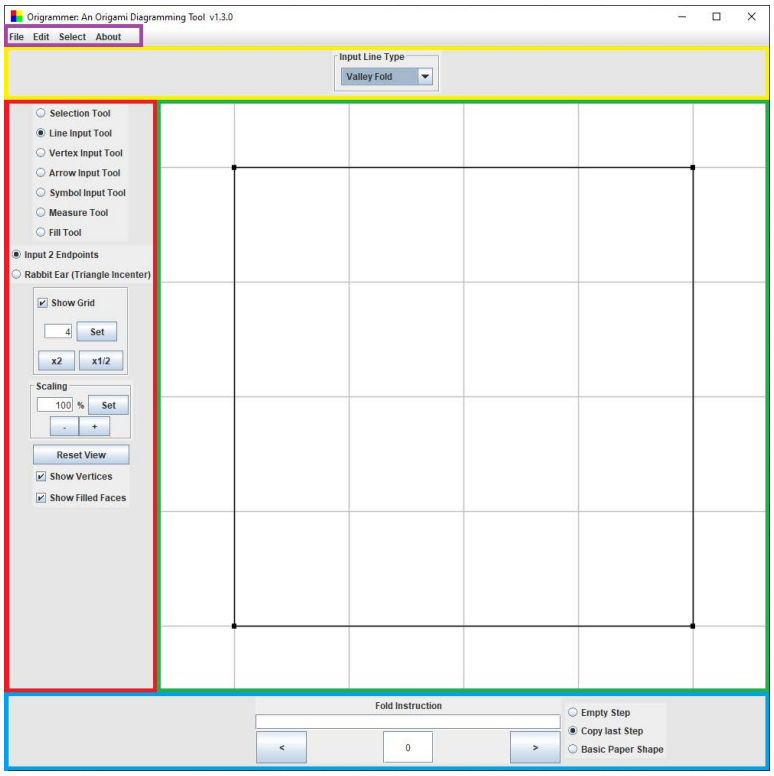


Figure 3.1: Menu Bar (Purple), Top Panel (Yellow), Side Panel (Red), Editing Panel (Green), Navigation Panel (Blue)

3.1 Menu Bar

The Menu Bar offers fast and easy access to diagram wide functions. Under `File`, the user can create a new diagram, open an already existing Origrammer diagram, save a diagram, and *export a diagram to a different format (not yet fully implemented)*. When creating a new diagram, the following panel appears, in which several settings for the diagram can be adjusted.

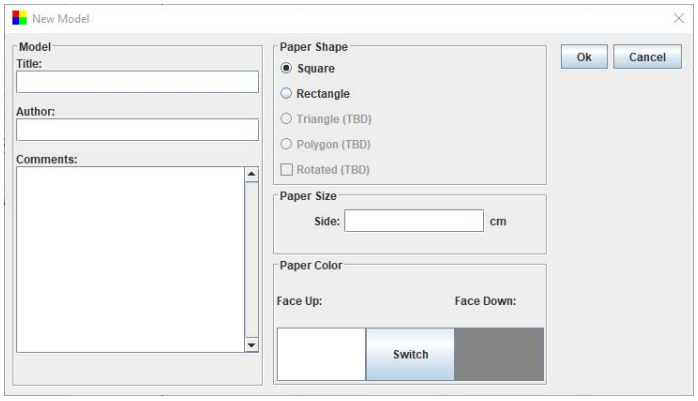


Figure 3.2: Options when creating a new diagram

The user is able to provide a `title`, an `author`, and `additional comments` (e.g. regarding copyright information). Additionally, paper related settings can be changed. The required paper shape can be defined (square, rectangle, triangle, polygon, or pre-rotated), a recommended paper size can be provided, and colours for both sides of the paper can be specified.

After confirming the settings for the new diagram, the paper outline of the first diagram step appears on the 3.2 Editing Panel.

The next button on the Menu Bar (`Edit`) includes the typical editing features `Undo` (Ctrl+Z), `Redo` (Ctrl+Y), `Cut` (Ctrl+X), `Copy` (Ctrl+C), `Paste` (Ctrl+V), and `Delete` selection (Delete). Additionally, the previously specified diagram information can be changed at any point under `Model Preferences` and some Origrammer-wide options can be adjusted under `Origrammer Preferences` (see Fig. 3.3).

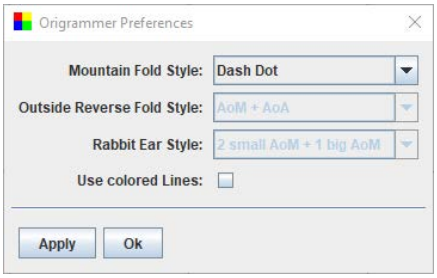


Figure 3.3: Origrammer Preferences

Lastly, under the `Select` button of the Menu Bar, the user can `Select all` (Ctrl+A) or `Unselect all` (Ctrl+Shift+A) objects on a given diagram step.

3.2 Editing Panel

The Editing Panel shows the current diagram step that is being worked on. All lines, arrows, and other symbols are being placed and edited here and all tools and input functions can then be used on the Editing Panel.

3.3 Side Panel

All major input tools and adjustments of the Editing Panel are displayed on the Side Panel. The input tools consist of the `Selection Tool`, the `Line Input`, `Vertex Input`, `Arrow Input`, `Symbol Input`, `Measure Tool`, and `Fill Tool`.

Through the `Selection Tool`, the user can select, move, and edit already placed objects on a diagram step. The `Line`, `Vertex`, `Arrow`, and `Symbol Input Tools` make it possible to place the respective objects on a given step. Furthermore, the user can measure the angle and length of lines within the diagram, in order to use these values as input for additional lines. The last tool is the `Filling Tool`, which offers the function to colour parts of the paper, according to the specified paper colour.

Furthermore, the options for the assisting grid of the Editing Panel can be changed, the scaling can be adjusted to see more detailed parts of a diagram step, and the rendering for vertices and the coloured faces of the diagram can be turned off.

3.4 Top Panel

After choosing a tool from the Side Panel, the respective input/editing options of the activated tool are being displayed on the Top Panel. These can be for example sliders for rotations, check boxes for mirroring a symbol, combo boxes for choosing different line/arrow types, and other things.

3.5 Navigation Panel

The Navigation Panel makes it possible to move through the individual diagram steps, as well as to create new ones. When creating a new step, the user has three different options. The new step can either be completely empty, only consist of the original paper outlines, or can be a simple copy of the previous step.

This is also the area, where the user can write down the textual folding instructions for a given step.

4 Selecting Evaluation Methods

4.1 10 Usability Heuristics by Jakob Nielsen and Rolf Molich

"A disadvantage of the method is that it sometimes identifies usability problems without providing direct suggestions for how to solve them."[6]

The Origrammer should mostly be tailored for professionals that are already well versed in the area of Origami. There is a lot of initial knowledge required when developing new Origami models and subseuently diagrams. This is the main reason why novices most likely won't be using the Origrammer. But that still means that knowledgeable Origami artists should be assisted where possible through Origrammers featureset.

5 Evaluating Origrammer

To begin the evaluation process the 10 usability heuristics by Jakob Nielsen [6] will be used in order to facilitate a base, on which can be build upon with other evaluation methods if required. As established in 4, the Usabilty Heuristics sometimes only identify problems without providing direct solutions. This is why this chapter will focus on finding problems and shortcomings of the Origrammer first. Afterwards, solutions can be developed that optimally fix most, or all, discovered issues without contradicting or counteracting other measures.

As the current (at the time of writing this thesis) Covid-19 pandemic hinders user involvement for the evaluation process, other measures have to be taken to ensure maximum efficiency and thoroughness. This is why there is an exhaustive list of all parts and features of the Origrammer below. Figure 5.1 roughly shows what features are located where on the Origrammer.

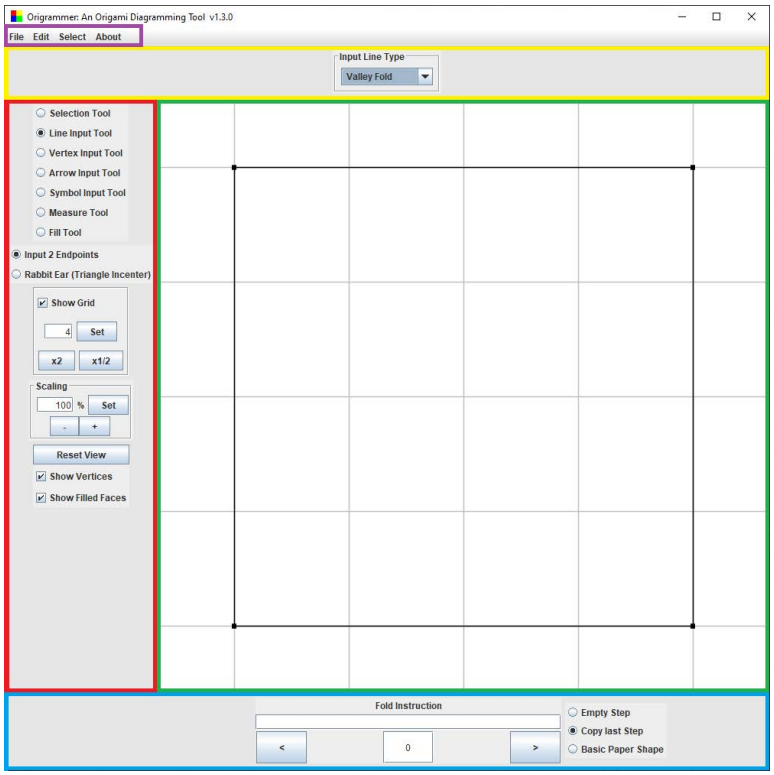


Figure 5.1: Menu Bar (Purple), Top Panel (Yellow), Side Panel (Red), Editing Panel (Green), Navigation Panel (Blue)

5.1 Origrammer Feature List

1. **Menu Bar (Purple)**
   (a) New File (b) Open File (c) Save File
   (d) Export File (e) Model Preferences (f) Origrammer Preferences
2. **Side Bar (Red)**
   (a) Selection Tool
      1. Click to Select 2. Hover over Object 3. Rectangular Selection
   (b) Line Input
      1. Two Point Input 2. Triangle Incenter
   (c) Vertex Input
      1. Absolute Position 2. Fraction of a Line
   (d) Arrow Input
      1. Valley Fold 2. Mountain Fold 3. Turn over
      4. Push Here 5. Pull out 6. Inflate here
   (e) Symbol Input
      1. Leader 2. Repetition Box 3. Next View Here
      4. Rotations 5. Hold Here 6. Hold Here and Pull
      7. X-Ray Circle *8. Fold Over & Over* 9. Equal Distances
      10. Equal Angles 11. Crimps 12. Pleats
      13. Closed Sinks
   (f) Measure Tool
      1. Measure Length 2. Measure Angle
   (g) Fill Tool
   (h) Grid Settings
   (i) Scaling Settings
3. **Navigation Panel (Blue)**
   1. Fold Instructions 2. Step Navigation 3. New Step Options
4. **Top Panel (Yellow)**
   See Side Panel for related features, as options for the selected Side Panel Tools appear on the Top Panel.
5. **Editing Panel (Green)**
   See Side Panel for related features, as the selected Side Panel Tools are being used on the Editing Panel.

5.2 10 Usability Heuristics

With the Origrammer Feature List as a basis, the evalutation using the 10 Usability Heuristics can be carried out. Every feature from the list will be checked against all 10 heuristics in order to try and maximise the completeness of the result. The found usability issues will then lead to the planning of potential usability improvements. Additional focus should be set on the 7th heuristic *Flexibility and Efficiency of Use*, as it includes the main goal of improving the efficiency of creating diagrams with the Origrammer.

1. Visibility of System Status

Nr:	Affects	Impact	Description
1.01	2.b; 2.d-2.g	3	When an action requires multiple input points (e.g. placing a Fold Line by two endpoints), the user doesn't know where he is in the process.
1.02	2.d; 2.e	7	Always show a preview of the `OriArrow` or `OriSymbol` before final placing
1.03	3.2; 3.3	6	The user doesn't know how many diagram steps there are overall and where they are

2. Aesthetic and Minimalist Design

Nr:	Affects	Impact	Description
2.01	2.a-2.i	4	The text in the Tool Selection on the Side Panel should be replaced by icons
2.02	4	3	The text in the TopPanel for input and editing options should be replaced by icons
2.03	2.f; 3	6	The User Interface should always fit properly (is currently not fitting in the SidePanel for MeasureTool & for some settings in the TopPanel)

3. User Control and Freedom

Nr:	Affects	Impact	Description
3.01	2.a; 2.b; 2.d; 2.e; 2.g	5	User should always be able to cancel an action (e.g. when inputting an object with multiple input points))
3.02	3.2; 3.3	8	Navigating through the diagram steps is unflexible and slow

4. Consistency and Standards

Nr:	Affects	Impact	Description
4.01	2.a; 2.b; 2.d; 2.e; 2.g	2	Editing options on the TopPanel should be labeled correctly (consistent naming scheme needed)
4.02	2.a; 4; 5	6	When selecting and editing Symbols/Arrows make it consistent (e.g. when selecting different OriObject types like OriArrows and OriSymbols at the same time, the TopPanel overfills)
4.03	1; 2; 3; 4	6	UI elements should have consistent sizes (e.g. "Set" Buttons, `JTextFields`)
4.04	1; 2; 3; 4	4	The sequential placement of UI parts should be consistent (e.g. for Input and Editing options on the TopPanel, have `JTextFields` first and then `JCheckBoxes` to the right

5. Error Prevention

Nr:	Affects	Impact	Description
5.01	2.b.2	7	Selecting the same point multiple times breaks the RabbitEar lines
5.02	2.e.10	8	Selecting points in the wrong order breaks inputs for the EqualAngle symbol
5.03	2.g	7	Selecting the same point multiple times breaks the FillTool
5.04	1.a; 1.e;	6	Restrict the input of every JTextField (e.g. only allow numbers for number inputs)

6. Recognition Rather than Recall

Nr:	Affects	Impact	Description
6.01	2.f	5	User should not be forced to remember the measured values (currently the measured values are being hidden, once the user selects a different tool from the SideBar)

7. Flexibility and Efficiency of Use

Nr:	Affects	Impact	Description
7.01	2.h	3	Give shortcuts for the Grid halve/ double buttons
7.02	2.i	3	Give shortcuts for the ResetView button
7.03	2.i	3	Give shortcuts for the step by step zoom-in/zoom-out
7.04	2.b	8	Give more Line Input options to avoid tedious work with grid adjustments
7.05	2.d; 3; 4	8	Give more flexibility when editing or placing arrows
7.06	2.e; 3; 4	8	Give more flexibility when editing or placing symbols
7.07	3.2; 3.3	9	The user can not change the order of steps, remove unwanted steps, or enter new steps between existing ones
7.08	2.g	7	The user has to select the vertices by hand and is limited to triangles
7.09	5	6	The user has to repeat similar diagram beginning steps for every new diagram.

8. Recognition, Diagnosis and Recovery from Errors

Nr:	Affects	Impact	Description
8.01	2.b-2.	5	Error messages for wrong user inputs should be self-explanatory
8.02	2.b-2.g	4	Error messages for wrong user inputs should explain how to fix the error

9. Help and Documentation

Nr:	Affects	Impact	Description
9.01	2.a-2.i; 3; 4	6	Show tooltips for all icons or non self-explanatory parts
9.02	2.a-2.g	7	Show short explanation on how inputs work for every possible input feature (could be combined with Nr.01 –> 5.2 Visibility

10. Match between System and Real World

Nr:	Affects	Impact	Description
10.01	2.b-2.e	5	Use "Origami terminology" everywhere (e.g. Rabbit Ear instead of Triangle Incenter; Next View Here Symbol instead of Eye Symbol)

6 Planning Solutions

This section will build on the discovered usability issues found during the initial evaluation process (see 5). On this basis, first solutions can be planned that fix the most impactful usability issues. Furthermore, new features can be developed that increase productivity (by automating processes and part of the workflow), while also fixing the current problems.

6.1 Graphical User Interface related changes

These changes include everything that is related to the graphical user interface that the user interacts with.

6.1.1 Overall UI Changes

Every feature area (e.g. grid options, scaling options, arrow/symbol inputs etc.) should be seperated through a `EtchedBorder` with its name as the title. Within these borders, elements should follow a consistent order. This is why this hierarchy is being set for the placement within these areas: `JRadioButton, JTextField, JButton, JCheckBox`.

Though additional `JLabels` can be used to explain or to give indicators what the elements do specifically. Whenever possible, icons should replace text and tooltips should be used to give further explanation. Explanatory `JLabels` should only be used when icons and tooltips are not enough.

Additionally to the more consistent placement of elements within the UI, their sizes should be standardized as well. `JTextFields` for example should only be large enough to encapsulate the largest possible entry. This means that the `height` should always be set to 25 pixels (and the `width` depending on what is being entered (e.g. entering an angle between 0-360° will be shorter than entering the textual explanation of a folding step).
Fixes: #2.01; #2.02; #2.03; #4.01; #4.03

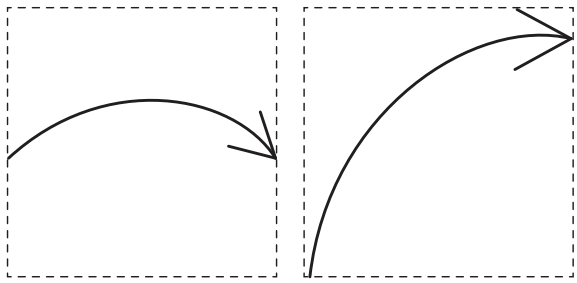
At a later point a complete redesign and rework of the UI will have to be carried out, in order to facilitate the goal of a unified and obstructionless design. At the current stage of development though the focus is being set to increase the efficiency and speed of creating origami diagrams, as this is the main reason why the Origrammer exists.

6.1.2 Arrows & Symbols

A large part of the Origrammer is made up of the different arrows and symbols that can be used in diagrams. So far, these objects were simple vector graphics that were being placed on the diagram as `ImageIcons` of `JLabels`. This approach did initially work, but brought restrictions and problems with it.

As vector graphics do not have explicit width or height values, the library Batik[1], which was being used to load the .svg files, presented wrong values to the `JLabel.setBounds(width, height)`-method. As a result of this limitation, the arrows & symbols got partially cut off at the original bounds of the `JLabel` when rotating them. To fix this issue, a new, pre-rotated vector graphic was being loaded whenever an arrow or symbol got rotated. Though this in turn facilitated itself in a wrong, always square border around the arrows and symbols. Additionally, this made interactions with arrows and symbols far slower and unflexible.

Another sideeffect was a change in scale when rotating a non square object. A `JLabel` tries to display the biggest possible object that can fit within the border bounds. As seen on Figure 6.1 the size of the arrow changes after rotating it by 45°.



(a) Straight Arrow (b) Angled Arrow

Figure 6.1: Unwanted Scaling when Rotating and the wrong, always square hitbox

As a result of all these problems and limitations, the decision was made to completely rework how arrows & symbols function in the Origrammer. The new approach was to rebuild all objects with simple `java.awt.Shape`-objects. This gave total control over rotations, scaling, and on-the-fly-editing of all arrows/symbols. Another advantage was the removed reliance on

`JLabels` and associated with that, there were no longer issues with wrong hitboxes or different scaling while rotating. The only disadvantage was the work-intensive nature of remodeling all arrows and symbols with `Shape`-objects by hand.

Fixes: #1.02; #7.05; #7.06

6.1.3 Filling Tool

The current limitations of the Filling Tool make it slow and cumbersome to use. The user has to manually input three vertices in order to create a triangle that shows the specified filling color. The limit of only three vertices per `OriFace`-object was originally introduced to give finer control for the user, as well as simplifying the development process for this feature.

But this decision went against the original goal of minimizing the time needed to create diagrams. As a result the Filling Tool has to be reworked, so that it can be used faster and with less user inputs.

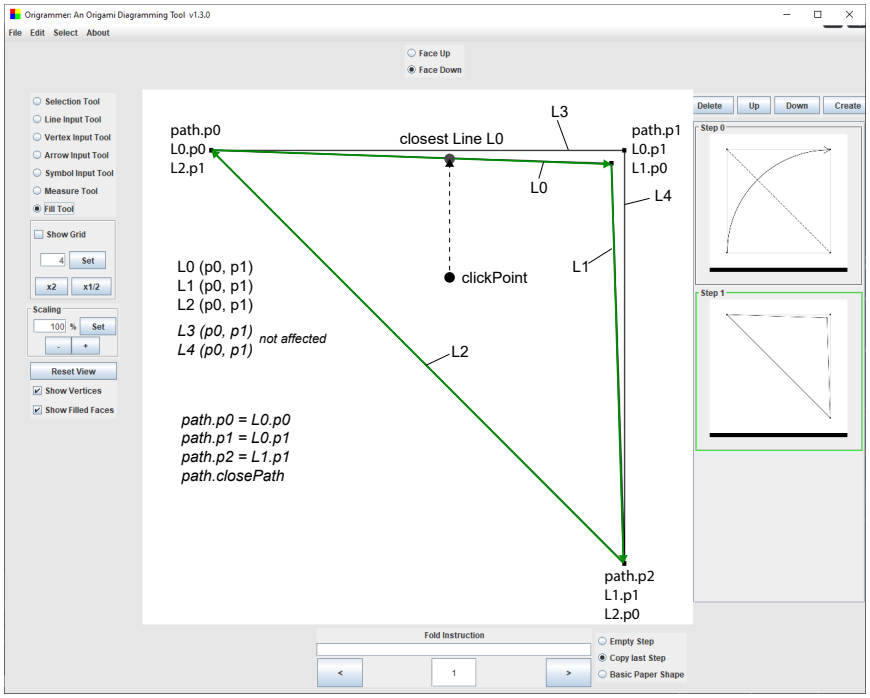


Figure 6.2: Example on how the Filling Tool works after the rework

The new Filling Tool only requires one mouse click to fill an area with colour. When clicking in an area, the Origrammer first looks for the closest

line to the `clickPoint`. Afterwards, both endpoints of the line are being checked for other touching or intersecting lines. The additional lines that were found have to be checked if they are visible from the original `clickPoint`, which would mean, that they are part of the enclosing lines. The endpoints of every new discovered line will be checked the same way, until a full enclosure of lines is established. Once this has happened, a `GeneralPath` is created that, goes through all points of the enclosing area. Lastly, the area of the `GeneralPath` can be filled with the specified colour.

6.1.4 Folding Presets

In origami diagramming, there are eight widely established bases[5] (p.53-64). When starting with these bases, one can fold a variety of different models. The bases in question are:

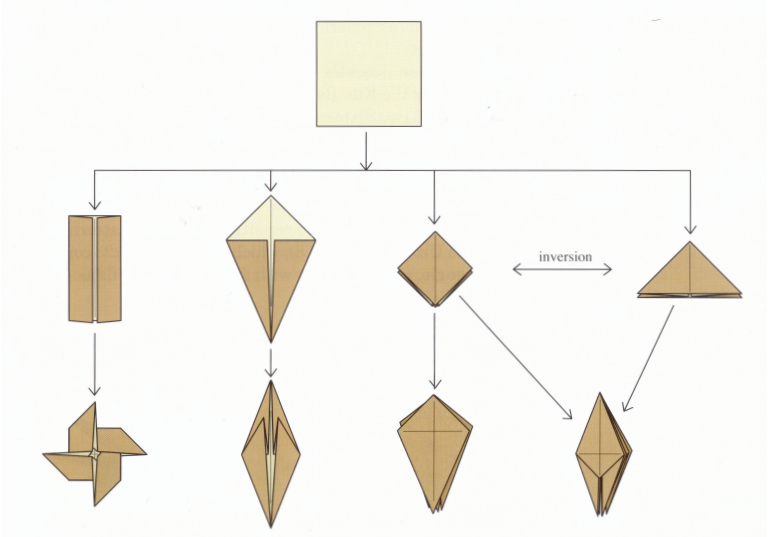


Figure 6.3: Family Tree of the standard bases.
Top to bottom and left to right: Cupboard Base, Windmill Base, Kite Base, Fish Base, Preliminary Fold, Bird Base, Waterbomb Base, Frog Base - [Robert J. Lang 2011 [4]]

As these bases serve as the starting point for a wide array of models, it would be helpful to include them in the Origrammer. By giving the user the option to start a diagram with one of these bases, the overall diagramming process will be shortened by quite a bit. Not having to keep repeating the same beginning steps for different models will save time and can also show beginners of the program, how parts of the Origrammer look like or work.

Additionally to the origami bases, instructions on how to fold different grid sizes, should be included in the folding presets as well. Folding a 3x3, 5x5, or 7x7 grid from a square paper is not as straight forward and requires multiple folding steps to achieve.

Fixes: #7.09

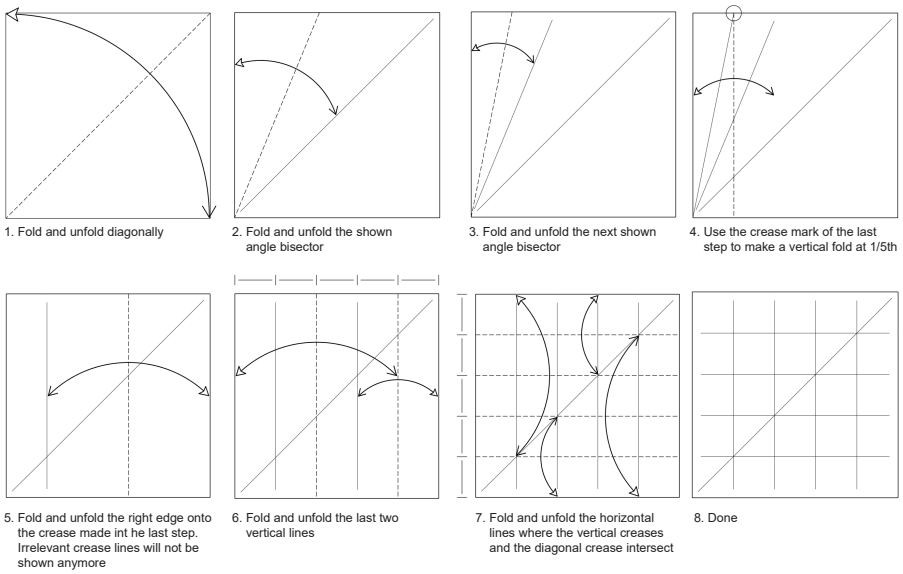


Figure 6.4: Example diagram of a 5x5 grid made with the Origrammer

6.1.5 Side Panel

The text of the `JRadioButtons` at the tool bar should be replaced by self-explanatory icons in order to improve overall clarity. But the textual explanation of all the tools should still be present to help especially novizes. This can be done through tooltips that appear when hovering over the buttons.

Fixes: #2.01; *#2.03*

6.2 User Input related changes

6.3 New Origrammer features

These new features should increase the productivity and efficiency of the Origrammer. The focus should be set on maximising the work that can be done in the shortest amount of time. This will be achieved by implementing features that either automate parts of the workflow, or that give new, faster possibilities of achieving the goal.

6.3.1 Step Navigation & Step Editing Options

Currently, the Origrammer can create new diagram steps with three different options (*Empty Step*, *Copy last Step*, and *Basic Paper Shape*). Furthermore

the user can navigate through them with a *Next-Step-/* and a *Previous-Step-*button. Though this approach brings limitation with it. Moving through the steps one by one hinders the overall usefulness and especially the speed of working with the Origrammer.

Deleting redundant steps can be quite important once the folding sequence goes through changes or simply when the user has made a mistake. In the same sense of offering solutions to user made mistakes, the Origrammer should make it possible to move steps to a different position within the diagram. Being able to create new steps in between existing ones can also be quite useful in negating user error.

An additional problem arises, once a diagram with a lot of steps (100+) is created. As the user currently only knows where he is in the folding sequence by the step number, this can lead to confusion. For example when folding an animal, there can be a different folding sequence, depending on what the artist decides to fold first (e.g. head first, the front or the hind legs, the tail, the abdomen, or the back). In order to continuously see where the user is within the diagram, there should be a small preview picture for all the steps within the step navigation. This will give the user an idea on what area a range of steps is working on. When clicking on one of the previews, the clicked step should made active and should be displayed on the Editing Panel.

As a result of the mentioned issues, the step navigation will be reworked to offer more functionality, visibility and ease of use. This will be accomplished by offering the following features in the rework:

- Small preview picture of a step
- Click on preview picture to show the step in the Editing Panel
- Show step number, the preview picture, as well as the folding description of the step in the navigation
- Delete a selected step
- Move a selected step to the front or the back
- Create a new step between existing steps

Fixes: #1.03; #3.02; #7.07

7 Problems & other Findings

This section discusses decisions and problems during the development process.

8 Prospect

Glossary

Origami (jpn: *ori* = *folding* and *kami* = *paper*) is the art of folding paper into models of animals, people or other objects. 5

Origrammer Origrammer is the name of the diagramming program developed within the scope of this project. 5

List of Figures

References

[1] The Apache Software Foundation. Apache™ batik svg toolkit, May 2020. https://xmlgraphics.apache.org/batik/ [Online; accessed July 07, 2020].

[2] Julian Hardtung. Origami diagramming: Development of a desktop application for creating origami diagrams, April 2020.

[3] Anrew Hudson. Crane, 2011. https://ahudsonorigami.files.wordpress.com/2011/11/tsuru.pdf [Online; accessed March 21, 2020].

[4] Robert J. Lang. Family tree of the standard bases. Published in: *Origami Design Secrets: mathematical methods for an ancient art*; p. 61.

[5] Robert James Lang. *Origami Design Secrets: mathematical methods for an ancient art*. CRC Press, second edition edition, 2012. ISBN: 978-1-56881-436-0.

[6] Jakob Nielsen. 10 usability heuristics for user interface design, April 1994. https://www.nngroup.com/articles/ten-usability-heuristics/ [Online; accessed June 10, 2020].

[7] Joseph Wu. Origami: A brief history of the ancient art of paperfolding, 2006. http://www.origami.as/Info/history.php [Online; accessed Juli 08, 2020.

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