FRAMESHIFT: SHIFT YOUR ATTENTION, SHIFT THE STORY

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Master of Science
in
Computer Science with a Concentration in Digital Arts

by

Tim Tregubov

in Conjunction with Rukmini Goswami

Department of Computer Science

DARTMOUTH COLLEGE

Hanover, New Hampshire

May 15, 2015

Examining Committee:

Chair ____________________
Lorie Loeb

Member ____________________
Michael Cohen

Member ____________________
Michael Casey

____
F. Jon Kull, Ph.D.
Dean of Graduate Studies

FRAMESHIFT: SHIFT YOUR ATTENTION, SHIFT THE STORY

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Master of Science
in
Computer Science with a Concentration in Digital Arts

by

Rukmini Goswami

in Conjunction with Tim Tregubov

Department of Computer Science

DARTMOUTH COLLEGE

Hanover, New Hampshire

May 15, 2015

Examining Committee:

Chair ____________________
Lorie Loeb

Member ____________________
Michael Cohen

Member ____________________
Michael Casey

____
F. Jon Kull, Ph.D.
Dean of Graduate Studies

ABSTRACT

INSERT ABSTRACT HERE

Acknowledgements

To our

To our families

Contents

I. TODO . . . 1
II. Tool Interactions . . . 3
(a) Box Fold . . . 3
(b) FreeForm . . . 3
(c) Polygon . . . 3
(d) V-Fold . . . 3
III. Interface Data Structures . . . 4
(a) Edges . . . 4
(b) Planes . . . 5
(c) Fold Features . . . 5
IV. Sketch . . . 6
V. TODOfds . . . 7
VI. TODO . . . 8
VII. TODO . . . 9
VIII. Nested Features . . . 10
IX. Deletion . . . 10
X. TODO . . . 11
XI. TODOsf . . . 12
XII. TODO23 . . . 13
XIII. User Test at the Digital Arts Exhibition . . . 14
XIV. Final User Study . . . 16
XV. TODO . . . 17
Appendix A: User Study 1 . . . 18
Appendix B: User Study 2 . . . 19
Appendix C: Distribution of Work . . . 20
Appendix D: Preview images from FrameShift — the novel . . . 21

List of Tables

List of Figures

I. TODO

fdsfsd

TODO: include preliminary user study 1

sfdaf

II. Tool Interactions

TODO: add tap options, how to draw, and a description of the tool-based interface in general

(a) Box Fold

To draw a

(b) FreeForm

(c) Polygon

(d) V-Fold

III. Interface Data Structures

We will refer to several data structures throughout the discussion of user interface design and implementation. These are the primary means of storing user input processed by the tools [1].

(a) Edges

An Edge represents a cut or fold. Edges are the basic building block of planes, and a basic element of . An edge is minimally defined by a start point, end point, and a a type (either cut or fold). This represents a straight edge between two points. In addition, an Edge can contain further information: the path drawn to create it, and a reference to the plane or feature it is a part of. Additionally, each edge contains a reference to its twin edge.

Twins

Although it is often simpler to think of edges as cuts and folds created by the user, the reality in Foldlings is slightly more complicated. For each edge that the user creates using a tool, two edges are created. One in the . Processing the edges. Having alternate orientations for edges also In order. Each edge is actually defined by two edges,

>>TODO: DIAGRAM

Driving Folds

A driving fold is not a special type of edge, but rather a relationship between an edge in one feature and a feature "spanning" that edge. A feature is said to span a fold when it is drawn on top of an existing fold.

>>TODO: boxfold diagram

[1] This section only describes the primary data structures necessary for, not structures for drawing features in 2D or 3D. For a discussion of 2D drawing, see sections. For a discussion of, see . **>>TODO:cite marissa >>TODO reference section**

An edge can be the driving fold for more than one feature, but each feature has only one driving fold (if there are multiple potential driving edges at the same height, the leftmost edge is selected. The driving fold is important for calculating parent-child relationships between features: a feature's parent is the feature that contains it's driving fold [2]. These parent-child relationships are described in more detail in the Nested Features section on page 10.

(b) Planes

A plane is a collection of Edges. **TODO: CITE MARISSA HERE**

(c) Fold Features

The central data structure of Foldlings is the FoldFeature: a representation of a shape drawn by the user that folds in 3d. Each fold feature is a single design element — and can be individually created, modified, and deleted. There are five types of FoldFeature: MasterCard, BoxFold, FreeForm, Polygon, and V-Fold, representing differences in drawing behavior, geometry, and (the differences are described in detail below). Each of these features is a subclass of the FoldFeature super class.

All FoldFeatures have functionality in common:

- Each feature contains a list of edges in the feature — both cuts and folds
- Each feature has a driving fold — in the case of unconnected features, such as the master card and holes, the driving fold is nil.
- Each feature can be deleted from the Sketch, "healing" the sketch by closing gaps left in any
- Features implement the encodeWithCoder and decodeWithCoder methods, allowing them to be serialized to a file on the device and restored from the saved file.

[2] The exception to this rule is holes — a hole's parent is the feature that contains it.

- Each feature can provide a list of current "tap options" — actions that can be performed on the feature given its state. **TODO:SEE tap options in interface design** [**?**]. **TODO:REMOVE – JUST TESTING**

Master Card

Each sketch always contains a single master feature, which is the ancestor of all

Box Fold

FreeForm

Holes are a special case of FreeForm shapes. FreeForm shapes that do not cross a fold are considered holes — drawn in white in the 2d sketch and drawn as subtractions from planes in the 3d view.

Polygon

V-Fold

IV. Sketch

A sketch is the representation of the user's drawing — it's primary role is as a collection of features. It also contains information about the current drawing state,

V. TODOfds

afsdfs dsf sf

VI. TODO

sfds

VII. TODO

sfds

VIII. Nested Features

a

IX. Deletion

a

X. TODO

sfsfs

XI. TODOsf

sfds

XII. TODO23

fdjas

XIII. User Test at the Digital Arts Exhibition

On April 28, 2015, we tested our system with attendees of the Digital Arts Exhibition at Dartmouth. After a brief demonstration of how to create folds and preview their design, users designed cards using Foldlings. Users drew sketches and then sent an email containing an SVG file to the computer connected to the laser cutter. Finally, they placed a piece of paper in the laser cutter, and watched as the laser beam cut out their design. Over the course of two hours, users cut and folded 31 popup cards. <<TOOO: CITE DAX <<TODO:

The system we demonstrated at the exhibition was incomplete — it contained the basic box fold and freeform shape tools, but did not include some advanced features of the final software, such as dragging folds or shading based on plane orientation. The alpha software also contained several bugs that disrupted the experience. However, the system was usable enough for people to create cards, and observing user behavior was invaluable in designing our final product.

Because users were new to our system — and constrained by the pressure of other users waiting to design cards — designs were relatively simple. Sketches generally contained between 2 and 5 fold features in addition to the base card — the most complex design contained 10 fold features. Despite their simplicity, sketches showed a wide range of designs, ranging from abstract shapes to representational scenes — users sketched symbols, Chinese characters, and geometric forms. Most of the sketches utilized both freeform and box fold features, mixing the two element types to create a composition. One of the most popular design elements was the user's name: 5 of the cards contained names or initials. Roughly a third of the designs took advantage of nesting — constructing fold features inside each other.

Because users were able to quickly design and fabricate their design, people generally left satisfied. People typically spent around 20 minutes at our booth, leaving with a popup card they had created. However, the experience was not frictionless. Users were frustrated by crashes: touching the screen with more than one finger or drawing while calculating planes were the most common reasons for failure. Other common complaints were the lack of a delete/undo button and that the UI did not show which tool was currently selected.

Folding the fabricated design also presented difficulties. Although they were able to see a 3d preview of their design while creating it, users had often relinquished the iPad by the time they folded their design. They were often unsure how to fold their card, and struggled to discover the correct fold orientations. In some cases, it took longer for users to fold their creation than to design it.

We observed several unexpected behaviors. A few users rotated the screen to design a card in a landscape view, rather than the portrait orientation implied by the orientation of the buttons and 3d preview. They used this orientation to design cards that folded medially rather than laterally. Several users also constructed overlapping features by drawing on top of existing features. These features did not simulate correctly, as they intersected with existing edges. However, this behavior demonstrated a desire to construct more complex geometry. In the final software we implement unions for fold features — the most recently-drawn feature occludes features underneath it, modifying their edges.

Users also relied on the 3d preview to differing degrees. Some users viewed the preview after every operation, while others only switched to the preview occasionally. Many users relied on the 3d preview as a reference to how to fold their popup card. We were surprised by this, and conducted further user studies to determine the effectiveness of methods of displaying 3d information.

XIV. Final User Study

As a final test of software, we compared the usability with Foldlings to traditional manual popup design methods. First, we presented participants with a completed popup card, and asked them to replicate the design using manual cutting and folding and to create the design using our software. The order was randomized, so half of participants were given were given the manual first, and half started with Foldlings. We timed how long the participants took to complete the design. Next, we gave them a free-form exercise: make a design that includes a tree." Finally, we asked participants to rate the experience of using our tool and their satisfaction with the popup card they produced. >>TODO: attach image >>TODO graphs, data, participants, do the study lol

>>TODO: update

XV. TODO

Since. >>TODO: side other research on visual aids for folding slash other 2d to 3d vis

goal: test whether users understand the mapping of 2d fold patterns to 3d, and test the degree to which plane coloring, edge patterning, and a 3d preview help users understand how a popup card will fold.

Each subject received a set of five laser cut cards, and we recorded the time it takes them to successfully fold the card.

For each card, each subject was randomly given one of of the following five aids:

1) A two-dimensional design, showing planes shaded by whether they will be horizontal or vertical when folded.
2) A two-dimensional design, with edges patterned based on whether they are “hills” or “valleys” — whether they fold towards or away from the card.
3) A video showing a simulation of the card folding in three dimensions.
4) A still image of the card folded in dimension.
5) No visual aid.

The order of aids was shuffled randomly, and then balanced to ensure an equal distribution of orderings. I.e. each

Finally, we asked subjects to rank the visual aids

The effect each type of aid has on folding time will help determine which types of visualization to include in Foldlings

Appendix A: User Study 1

Appendix B: User Study 2

Appendix C: Distribution of Work

Appendix D: Preview images from FrameShift the novel