DEUCE
A Lightweight UI For Structured Editing

Brian Hempel, Justin Lubin, Grace Lu, and Ravi Chugh
Unstructured text is clumsy
GHOST: So art thou to revenge, when thou shalt hear.

HAMLET: What?

GHOST: I am thy father's spirit,
Doom'd for a certain term to walk the night,
And for the day confin'd to fast in fires,
Till the foul crimes done in my days of nature
Are burnt and purged away. But that I am forbid
To tell the secrets of my prison-house,
I could a tale unfold whose lightest word
Would harrow up thy soul; freeze thy young blood,
GHOST: So art thou to revenge, when thou shalt hear.

HAMLET: What?

GHOST: I am thy father's spirit,
   Doom'd for a certain term to walk the night,
   And for the day confin'd to fast in fires,
   Till the foul crimes done in my days of nature
   Are burnt and purged away. But that I am forbid
   To tell the secrets of my prison-house,
   I could a tale unfold whose lightest word
   Would harrow up thy soul; freeze thine young blood,
GHOST: So art thou to revenge, when thou shalt hear.

HAMLET: What?

GHOST: I am thy father’s spirit,
Doom’d for a certain term to walk the night,
And for the day confin’d to fast in fires,
Till the foul crimes done in my days of nature
Are burnt and purged away. But that I am forbid
To tell the secrets of my prison-house,
I could a tale unfold whose lightest word
Would harrow up thy soul; freeze thy young blood,

```
maybeZip : List a -> List b -> Maybe (List (a,b))
maybeZip xs ys = case (xs, ys) of
  (x::xs_, y::ys_) -> case maybeZip xs_ ys_ of
    Nothing       -> Nothing
    Just xys       -> Just (x,y) :: xys
  ([] , [])       -> Just []
  _               -> Nothing
```

→ Not a program
GHOST: So art thou to revenge, when thou shalt hear.

HAMLET: What?

GHOST: I am thy father’s spirit,
Doom’d for a certain term to walk the night,
And for the day confin’d to fast in fires,
Till the foul crimes done in my days of nature
Are burnt and purged away. But that I am forbid
To tell the secrets of my prison-house,
I could a tale unfold whose lightest word
Would harrow up thy soul; freeze thy young blood,

```haskell
maybeZip : List a -> List b -> Maybe (List (a,b))
maybeZip xs ys = case (xs, ys) of
  (x::xs_, y::ys_) -> case maybeZip xs_ ys_ o
    Nothing -> Nothing
    Just xys -> Just ((x,y):: xys)
  ( [], [] ) -> Just []
_ -> Nothing
```
GHOST: So art thou to revenge, when thou shalt hear.

HAMLET: What?

GHOST: I am thy father's spirit,
Doom'd for a certain term to walk the night,
And for the day confin'd to fast in fires,
Till the foul crimes done in my days of nature
Are burnt and purged away. But that I am forbid
To tell the secrets of my prison-house,
I could a tale unfold whose lightest word
Would harrow up thy soul; freeze thy young blood,

maybeZip : List a -> List b -> Maybe (List (a,b))
maybeZip xs ys = case (xs, ys) of
  (x::xs_, y::ys_) -> case maybeZip xs_ ys_ of
    Nothing    -> Nothing
    Just xys   -> Just ((x,y) :: xys)
  ([], [])    -> Just []
  _           -> Nothing
Problem for Beginners
Problem for Beginners

Arrows are reserved for cases and anonymous functions. Maybe you want > or >= instead?

46| Nothing --> Nothing
Structure must come from your head!

Arrows are reserved for cases and anonymous functions. Maybe you want > or >= instead?

46| Nothing -> Nothing
Problem for Experts
Problem for Experts

You

You

You

You

You

You

You

You

You

You

You

You

You

You

You

You

You

You

You

You

You

You
Problem for Experts

Spend time herding text!
Problem for Experts

Spend time herding text!
Problem for Experts

let x = 6 in case x of You

Spend time herding text!
Your program isn’t text.
Your program isn’t text.

It’s an AST.
Text Changes
Text Changes ≠
Text Changes \neq AST Changes
Structured Editors
Structured Editors

Scratch

(Maloney et al. 2010; Resnick et al. 2009)
Structured Editors

**Scratch**
(Maloney et al. 2010; Resnick et al. 2009)

**TouchDevelop**
(Tillmann et al. 2012)
Structured Editors

Scratch
(Maloney et al. 2010; Resnick et al. 2009)

TouchDevelop
(Tillmann et al. 2012)

UI challenges; Experts still use plain text
Traditional Refactoring
Traditional Refactoring

Text-Select    Menu    Configure
Traditional Refactoring

Text-Select

Menu

Configure

```java
@override
public ConvexOptimizer getOptimizer() {
    if (optimizer == null) {
        Solver solver = new Solver.Builder().model(this).configure(conf()).build();
        this.optimizer = solver.getOptimizer();
    }
    return optimizer;
}

/**
 * Returns the parameters of the neural network as a flattened row vector
 * @return the parameters of the neural network
 */
@override
public INDArray params() {
    return paramsFlattened;
}

@override
public INDArray getParam(String param) {
    return params.get(param);
}
```
Traditional Refactoring

Text-Select   Menu   Configure

@Override
public ConvexOptimizer getOptimizer() {
    if (optimizer == null) {
        return null;
    }
    return optimizer;
}

/**
 * @return convolution parameters
 * @param param a flattened row vector
 */

@Nullable
public float[] getParameters() {
    return parameters.get(param);
}
Traditional Refactoring

Text-Select   Menu   Configure
Traditional Refactoring

Text-Select  Menu  Configure
Traditional Refactoring

Text-Select  Menu  Configure

✗ Awkward
Traditional Refactoring

Text-Select  Menu  Configure

Extract Method

Not all selected statements are enclosed by the same parent statement.

OK
Traditional Refactoring

Text-Select  Menu  Configure

✗ Awkward
Traditional Refactoring

Text-Select    Menu    Configure

✗ Awkward
✗ Multiple Selections
Traditional Refactoring

- Text-Select
  - ✗ Awkward
  - ✗ Multiple Selections

- Menu
  - Extract Method...
    - Extract Interface...
    - Extract Superclass...
    - Use Supertype Where Possible...
    - Pull Up...
    - Push Down...
    - Extract Class...
    - Introduce Parameter Object...

- Configure
Traditional Refactoring

Text-Select
- Awkward
- Multiple Selections

Menu
- Many Options

Configure
Traditional Refactoring

✗ Awkward Text-Select Menu

Many Options

✗ Multiple Selections

Extract Method

Method name:

Access modifier: public protected package private

☐ Declare thrown runtime exceptions

☐ Generate method comment

☐ Replace additional occurrences of statements with method

Method signature preview:

private void extracted()
Traditional Refactoring

Text-Select
- Awkward
- Multiple Selections

Menu
- Many Options

Configure
- Dialogs
Traditional Refactoring

Text-Select
- Awkward
- Multiple Selections

Menu
- Many Options

Configure
- Dialogs
Deuce

Text-Select
- Awkward
- Multiple Selections

Menu
- Many Options

Configure
- Dialogs
(def image1
  (let [width height]
    (let [x y] [50 65]
      (image "lightgrey")
    (draw (concat [ image

Structure
Select

Menu
Many
Options

Configure
Dialogs

Deuce
(def image1
  (let [width height][50 65]
    (image "lightgrey")
)
(def main
  (draw (concat [ image1 ])))

Deuce

Structure Select

Short Menu

Configure

✗ Dialogs

✗ Dialogs
(def image1
  (let [\text{width} \text{height}]
    (let [x y] [50 65]
      (image "lightgrey")))

(def main
  (draw (concat [\text{image1}]))

\textbf{Deuce}

\textbf{Structure Select}

\textbf{Short Menu}

\textbf{Defaults}

- Abstract image1 over its constants
- Abstract image1 over its named constants
(def image_url "img/icse-2018-large-icon-small.png")

(def image1 (image "lightgrey" 50 65 283 254 15 image_url))

(def main (draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small.png")

(def image1
  (image "lightgrey" 50 65 283 254 15 image_url))

(def main
  (draw (concat [ image1 ])))
(def image-url "img/icse-2018-large-icon-small.png")

(def image1 (image "lightgrey" 50 65 283 254 15f image-url))

(def main)
(draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small.png")

(def image1
  (image "lightgrey" 50 65 283 254 15 image_url))

(def main
  (draw (concat [ image1 ])))
(def image_url "img/icse-2018-large-icon-small.png")

(def image1 (image "lightgrey" 50 65 283 254 15 image_url))

(def main (draw (concat [ image1 ])))
(def image_url "img/icse-2018-large-icon-small.png")

(def image1 (image "lightgrey" 50 65 283 15 image_url))

(def main (draw (concat [ image1 ])))
(def image_url "img/icse-2018-large-icon-small.png")

(def image1 (image "lightgrey" 50 65 283 254 15 image_url))

(def main (draw (concat [ image1 ])))
(def image_url "img/icse-2018-large-icon-small.png")

(def image1 (image "light-grey" 50 65 283 254 15 image_url))

(def main (draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small.png")

(def image1
  (image "lightgrey" 50 65 283 254 15 image_url))

(def main
  (draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small.png")

(def image1
  (image "lightgrey" 50 65 283 254 15 image_url))

(def main
  (draw (concat [ image1 ])))
(def image_url "img/icse-2018-large-icon-small.png")

(def image1 (image "lightgrey" 50 65 283 254 15 image_url))

(def main)

(draw (concat [ image1 ]))
(def image-url "img/icse-2018-large-icon-small.png")
(def image1 (image "lightgrey" 150 65 253 15 image-url))
(def main (draw (concat [ image1 ])))
(def image-url
  "img/icse-2018-large-icon-small.png")

(def image1
  (image "lightgrey" 50 65 283 254 15 image-url))

(def main
  (draw (concat [ image1 ])))
(def image_url "img/icse-2018-large-icon")
(def image1 (image "lightgrey" 50 65 283 254 15 image_url))
(def main (draw (concat [ image1 ])))
(def image_url "img/icse-2018-large-icon")
(def image1 (image "lightgrey" 50 65 283 254 15 image_url))
(def main (draw (concat [ image1 ])))
(def image-url "img/icse-2018-large-icon"
  (let width 283
    (image "lightgrey" 50 65 width 254 15 image-url)))
(def main
  (draw (concat [ image1 ])))
(def image_url  
  "img/icse-2018-large-icon-small.png")

(def image1  
  (let width 283  
    (image "lightgrey" 50 65 width 254 15 image_url))))

(def main  
  (draw (concat [ image1 ])))
(def image_url "img/icse-2018-large-icon-small.png")

(let width 283
 (image "lightgrey" 50 65 width 254 15 image_url))

(def main
 (draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small.png")

(def image1
  (let width 283
    (image "lightgrey" 50 65 width 254 15 image_url)))

(def main
  (draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small."

(def image1
  (let width 283
    (image "lightgrey" 50 65 width 264 15 image_url)))

(def main
  (draw (concat [ image1 ])))
(def image_url "img/icse-2018-large-icon-small."
  (image "lightgrey" 50 65 width 254 15 image_url)))
(def main
  (draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small"
)

(def image1
  (let width 283
    (let height 254
      (image "lightgrey" 50 65 width height 15 image_url)))
)

(def main
  (draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small.png")

(def image1
  (let width 283
    (let height 254
      (image "lightgrey" 50 65 width height 15 image_url)))
  
(def main
  (draw (concat [ image1 ]))
(def image_url
  "img/icse-2018-large-icon-small.png")

(def image1
  (let width 283
      (let height 254
          (image "lightgrey" 50 65 width height 15 image_url)))

(def main
  (draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small.png")
(def image1
  (let width 283
    (let height 254
      (image "lightgrey" 65 width height 15 image_url)))
(def main
  (draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-

(def image1
  (let width 283
    (let height 254
      (image "lightgrey" width height 15 image_url)))

(def main
  (draw (concat [ image1 ])))
(def image_url "img/icse-2018-large-"
  (let width 283
    (let height 254
      (let [x y [50 65]
        (image "lightgrey" x y width height 15 image_url)))))
(def main
  (draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small.png")

(def image1
  (let [width 283
        height 254]
    (let [x y [50 65]]
      (image "lightgrey" x y width height 15 image_url)))))

(def main
  (draw (concat [ image1 ])))
(def img/ico
   "img/ico-small.png")

(let [width 283
      height 254
      [x y] [50 65]
      (image "lightgrey" x y width height 15 image_url)))

(def main)
(draw (concat [ img1 ]))
(def image "img/icon"
   "lightgrey"
   [x y] [50 65]
   (image "lightgrey" x y width height 15 image_url)))

(def main
   (draw (concat [ image1 ])))
(def image "img/icon.png")

(def image-small "img/icon-small.png")

(let [width 283]
  (let [height 254]
    (let [xy [50 65]]
      (image "lightgrey" x y width height 15 image-url))))

(def main)
(draw (concat [ image1 ]))
(def image "img/icon.png")
(let [width height] [283 254]
  (let [x y] [50 65]
    (image "lightgrey" x y width height 15 image_url)))
(def main)
(draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small.png")

(def image1
  (let [width height [283 254]
        [x y] [50 65]
        (image "lightgrey" x y width height 15 image_url)))))

def main
  (draw (concat [ image1 ])))
(def image (fn [x y width height] (image "lightgrey" x y width height 15 image_url))))
(def main)
(draw (concat [ image1 ])))
(def image "img/ics")
(let [x y width height] [50 65 283 254]
  (image "lightgrey" x y width height 15 image_url)))
(def main
  (draw (concat [ image1 ])))
(def image_url
  "img/icse-2018-large-icon-small.png")

(def image1
  (let [x y width height [50 65 283 254]]
    (image "lightgrey" x y width height 15 image_url)))

(def main
  (draw (concat [ image1 ])))
(def image1
  (let [x y width height [50 65 283 254]
    (image "lightgrey" x y width height 15 image_url))]
(def main
  (draw (concat [ image1 ]))))
(def image1
  (let [x y width height [50 65 283 254]
        (image "lightgrey" x y width height 15 image_url))

(def main
  (draw (concat [ image1 ])))
(def image1 (\(x y width height bgColor padding) 
  (image bgColor x y width height padding image_url)))

(def main

  (draw (concat [ (image1 50 65 283 254 "lightgrey" 15) ])}
(def image1 \(x y width height\)
(image "lightgrey" x y width height 15 |image_url)))

(def main
(draw (concat [ (image1 50 65 283 254) ])))
(def image_url "img/icse-2018-large-icon-small.png")

(def image1 (\(x y width height\)
 (image "lightgrey" x y width height 15 image_url)))

(def main)
(draw (concat [ (image1 50 65 283 254) ])))
(def image_url "img/icse-2018-large"
  (image "lightgrey" x y width height 15 image_url)))

(def main
  (draw (concat [(image1 50 65 283 254)])))
(def image_url "img/icse-2018-large")
(def image1 (\(x y\) (image "lightgrey" x y 283 254 15 image_url)))
(def main (draw (concat [(image1 50 65)])))
(def image_url
"img/icse-2018-large-icon-small.png")

(def image1 (\(x y\)
(image "lightgrey" x y 283 254 15 image_url)))

(def main
(draw (concat [(image1 50 65)])))
(def image1 \(x y\)
     (image "lightgrey" x y 283 254 15 image_url)))
(def main
 (draw (concat [(image1 50 65)])))
(def image1 \((x y)\) (image "lightgrey" x y 283 254 15 image_url)))

(def main)
(draw (concat [(image1 50 65)]))
(def image1 (\(\(x y\) \\
  (image "lightgrey" x y 283 254 15 image_url))))

(def main 
  (draw (concat [ (image1 50 65) ]))))
(def image_url "img/icse-2018-large-icon-small.png")

(def icse2018 (\(x y\) (image "lightgrey" x y 283 254 15 image_url)))

(def main)
(draw (concat [(icse2018 50 65)]))
(def image_url "img/icse-2018-large-icon-small.png")

(def icse2018 (\(x y\) (image "lightgrey" x y 283 254 15 image_url)))

(def main)
(draw (concat [(icse2018 50 65)]))
Deuce

Structure
Select

Short
Menu

Defaults
Deuce

Structure Select

Short Menu

Defaults

Code Tools
Create Function from Definition
Inline Definition
Make Single Line

Code Tools
Move Definition
Duplicate Definition
Deuce

Structure
Select

Short
Menu

Defaults

Code Tools
Create Function from Definition
Inline Definition
Make Single Line

Abstract image1 over its constants
Abstract image1 over its named constants

Move Definition
Duplicate Definition

Move width and height
Deuce more effective than Traditional?
Deuce more effective than Traditional?

Deuce preferred to Traditional?
Deuce
Structure Select

```
(def image1
  (let [width height]
    (let [x y] [100 100]
      (image "lightgrey")
    ))
```

Deuce

Short Menu

- Move Definition
- Duplicate Definition

Defaults

- Move width and height
Deuce “Box-Select Mode”

Structure Select

```
(def image1
  (let [width height]
    (let [x y [100 100]
          image "lightgrey"]
      ...
```
Deuce “Box-Select Mode”

Structure Select
(def image1
  (let [width height]
    (let [x y] [100 100]
      (image "lightgrey")
    )
  )

Short Menu

Code Tools
- Move Definition
- Duplicate Definition

Defaults
- Move width and height
Traditional “Text-Select Mode”

Deuce “Box-Select Mode”

Structure Select

(def image1
(let [width height]
(let [x y] [100 100]
(image "lightgrey")

Short Menu

Code Tools
- Move Definition
- Duplicate Definition

Move width and height

Defaults
Traditional “Text-Select Mode”

Text Select

```
(def image1
  (let [width height]
    (let [x y] [100 100]
      (image "lightgrey"))
```

Deuce “Box-Select Mode”

Structure Select

```
(def image1
  (let [width height]
    (let [x y] [100 100]
      (image "lightgrey"))
```

Short Menu

- Move Definition
- Duplicate Definition

Defaults

- Move width and height
Traditional “Text-Select Mode”

Text Select

Right-Click Menu

Deuce “Box-Select Mode”

Structure Select

Short Menu

Defaults

```
(def image1
(let [width height]
(let [x y] [100 100]
(image "lightgrey")
```

```

(def image1
(let [\textit{width} height]
(let [x y] [100 100]
(image "lightgrey")
```
Traditional “Text-Select Mode”

Text Select

(def image1
  (let [width height]
    (let [x y] [100 100]
      (image "lightgrey")
  
Right-Click Menu

Select Arguments

(def image1
  (let [width height]
    (let [x y] [100 100]
      (image "lightgrey")

Deuce “Box-Select Mode”

Structure Select

(def image1
  (let [width height]
    (let [x y] [100 100]
      (image "lightgrey")

Short Menu

Defaults

Code Tools

- Move Definition
  - Select one or more variable definitions and one target position (i.e. whitespace)

Move width and height

Duplicate Definition
Traditional “Text-Select Mode”

Text Select

(def image1
 (let [width height]
  (let [x y] [100 100]
   (image "lightgrey")

Right-Click Menu

Select Arguments

Defaults

Deuce “Box-Select Mode”

Structure Select

(def image1
 (let [width height]
  (let [x y] [100 100]
   (image "lightgrey")

Short Menu

Code Tools

- Move Definition
- Duplicate Definition

Defaults

Move Definition

Requirements
- Select one or more variable definitions and one target position (i.e. whitespace) (Satisfied)

Code Updates

Move width and height
Tutorial
Tutorial

Head-to-Head Tasks (2x; once per mode)
Tutorial

**Head-to-Head Tasks** (2x; once per mode)

**Mix & Match Tasks** (free to use both modes)
Tutorial

**Head-to-Head Tasks** (2x; once per mode)

**Mix & Match Tasks** (free to use both modes)

Exit Survey
Deuce more effective than Traditional?
Deuce more effective than Traditional?
Deuce more effective than Traditional?
**Deuce** more effective than **Traditional**?
Deuce more effective than Traditional?
Deuce more effective than Traditional?
Deuce more effective than Traditional?

![Bar chart comparison between First and Second Encounter Completion Rates]

- First Encounter: [Completion Rate Bar]
- Second Encounter: [Completion Rate Bar]

Both Similar, p=0.17
Deuce more effective than Traditional?
**Deuce** more effective than **Traditional**?

Deuce doesn’t help discoverability
Deuce more effective than Traditional?

Rel Time (vs task mean)

4 -
3 -
2 -
1 -
0 -
Deuce more effective than Traditional?

Both Similar
p=0.52
Deuce more effective than Traditional?
**Deuce** more effective than **Traditional**?

Deuce 36% faster  
$p<0.01$
Deuce more effective than Traditional?

Deuce may be faster once learned

Deuce 36% faster  
$p<0.01$
Deuce preferred to Traditional?
Deuce preferred to Traditional?

Survey
Deuce preferred to Traditional?

Survey

#Responses

0 -
5 -
10 -
15 -

T  D
Deuce preferred to Traditional?

Survey
Deuce preferred to Traditional?

Survey

Modest subjective preference for Deuce
Deuce preferred to Traditional?

Survey:

Modest subjective preference for Deuce

Observed:

Survey Observed
Deuce preferred to Traditional?

Survey

Modest subjective preference for Deuce

Observed
Deuce preferred to Traditional?

Survey

Modest subjective preference for Deuce

Observed

Almost everyone used Deuce more
Mix & Match Tool Usage
Mix & Match Tool Usage

- Rename
- Make Equal with Single Variable
- Introduce Variable(s)
- Add Argument(s)
- Create Function from Arguments
- Move Definition(s)
- Inline Definition(s)
- Create Function by Merging Definitions
- Create Function from Definition
Mix & Match Tool Usage

- Rename
- Make Equal with Single Variable
- Introduce Variable(s)
- Add Argument(s)
- Create Function from Arguments
- Move Definition(s)
- Inline Definition(s)
- Create Function by Merging Definitions
- Create Function from Definition

Mean Proportion of User's Refactorings
Mix & Match Tool Usage

- Rename
- Make Equal with Single Variable
- Introduce Variable(s)
- Add Argument(s)
- Create Function from Arguments
- Move Definition(s)
- Inline Definition(s)
- Create Function by Merging Definitions
- Create Function from Definition

Mean Proportion of User's Refactorings

0% 5% 10% 15%
Mix & Match Tool Usage

- **Rename**
- **Make Equal with Single Variable**
- **Introduce Variable(s)**
- **Add Argument(s)**
- **Create Function from Arguments**
- **Move Definition(s)**
- **Inline Definition(s)**
- **Create Function by Merging Definitions**
- **Create Function from Definition**

*Mean Proportion of User's Refactorings*
Mix & Match Tool Usage

Hypothesis: **Deuce** better for multi-argument transformations

- Single argument transform

![Bar chart showing mean proportion of user's refactorings](chart.png)
Deuce vs Traditional
Deuce vs Traditional

Traditional may be better for learning
Deuce vs Traditional

Traditional may be better for learning

Deuce may be faster once learned
Deuce vs Traditional

Traditional may be better for learning

Deuce may be faster once learned

Deuce strongly preferred
Future Work
Future Work

UI concerns for larger programs
Future Work

UI concerns for larger programs

How to encourage refactoring?
Future Work

UI concerns for larger programs

How to encourage refactoring?

DSL for defining new transformations
Future Work

UI concerns for larger programs

How to encourage refactoring?

DSL for defining new transformations

Real languages in existing editors
Related Work
Related Work

Selection Assist + Box View
(Murphy-Hill and Black 2008)
Related Work

Selection Assist + Box View
(Murphy-Hill and Black 2008)

DNDRefactoring
(Lee et al. 2013)
Related Work

Selection Assist + Box View
(Murphy-Hill and Black 2008)

DNDRefactoring
(Lee et al. 2013)

Greenfoot
(Brown et al. 2016)

Barista
(Ko and Myers 2006)

Graphite
(Omar et al. 2012)
Structure
Select

(def image-url
 "img/icse-2018-large-icon-small.png")

(def image1
(let [width-height [324 200]
 (let [x y [100 100]
   (image "lightgrey" x y width-height 15 image-url)]))

(def main
 (draw (concat [ image1 ])))
Structure
Select

Short Menu

```
(def image-url
  "img/icse-2018-large-icon-small.png")
(def image1
  (let [width height [324 200]
        [x y] [100 100]
        (image "lightgrey" x y width height 15 image-url))
(def main
  (draw (concat [ image1 ])))
```

**Code Tools**
- Create Function from Definition
- Inline Definition
- Make Single Line
Structure
Select

Short
Menu

Defaults

Code Tools
- Create Function from Definition
- Inline Definition
- Make Single Line
- Abstract image1 over its constants
- Abstract image1 over its named constants
Deuce provides \textit{streamlined structural editing} in a familiar \textit{text-based} environment.
Search “sketch n sketch” to play with Deuce
Thank you!

Structure
Select

Short Menu

Defaults

Code Tools
- Create Function from Definition
- Inline Definition
- Make Single Line

- Abstract image1 over its constants
- Abstract image1 over its named constants

Search “sketch n sketch” to play with Deuce
Extra Slides
In this section, we explain the design of tool to move the de
ations through 9, as well as the space on line 13, and selects the
all design parameters and shapes within the single
program is a sequence of top-level de
itions (both at the top-level and locally via
transformations that are provided in our current implementation.
In this section, we explain the design of
tool to move the de
ations based on the set of structural se
ctions in the previous program, but that order was unintuitive—the
arguments to the function match the order of de
itions
Figure 1: Syntax of Little. The orange boxes and blue dots identify features for structural selection.
Figure 2: Code tool interface.
Widgets when hovering over the code box only when the user is text-editing mode. This allows the user to quickly toggle between Elm and JavaScript code. The new version (language in which programs are compiled to JavaScript and run in the step towards the goal of by heuristics; S

Figure 3: Example target positions.
<table>
<thead>
<tr>
<th>Code Tools</th>
<th>Code Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Function from Definition...</td>
<td>Make Equal with Single Variable...</td>
</tr>
<tr>
<td>Create Function from Arguments...</td>
<td>Make Equal by Copying...</td>
</tr>
<tr>
<td>Create Function by Merging Definitions...</td>
<td>Move Definition...</td>
</tr>
<tr>
<td>Add Argument...</td>
<td>Swap Definitions...</td>
</tr>
<tr>
<td>Remove Argument...</td>
<td>Inline Definition...</td>
</tr>
<tr>
<td>Reorder Arguments...</td>
<td>Duplicate Definition...</td>
</tr>
<tr>
<td>Rename Variable...</td>
<td>Reorder Expressions...</td>
</tr>
<tr>
<td>Introduce Local Variable...</td>
<td>Swap Expressions...</td>
</tr>
<tr>
<td>Swap Variable Names and Usages...</td>
<td>Make Single Line...</td>
</tr>
<tr>
<td>Swap Variable Usages...</td>
<td>Make Multi-line...</td>
</tr>
<tr>
<td></td>
<td>Align Expressions...</td>
</tr>
</tbody>
</table>
Head-to-Head Tasks

Mixed effects model to “control” for:
- participant skill
- trial number
- first/second encounter
- mouse/trackpad
- own/our computer
Head-to-Head Tasks

Mixed effects model to “control” for:

- participant skill
- trial number
- first/second encounter
- mouse/trackpad
- own/our computer

$p = 0.057$
Head-to-Head Tasks

Mixed effects model to “control” for:

- participant skill
- trial number
- first/second encounter
- mouse/trackpad
- own/our computer
Is either mode more effective for rapid editing?

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.

Among trials for which a transformation was performed, Traditional Mode was better when they knew the desired transformation. How-ever, to tease out if any of these differences in user skill and task difficulty per task. Reported p-values are based on Wald Z-statistics.
Head-to-Head Tasks

Figure 4: Task completion rates pooled over both modes.
Head-to-Head Tasks

“Which interaction worked better for the … task?”

Modest subjective preference for Deuce
Mix & Match Tasks

What did participants actually use?

Deuce preferred by almost all users.