Sketch-n-Sketch: Interactive SVG Programming with Direct Manipulation

Ravi Chugh, Brian Hempel, Mitchell Spradlin, Jacob Albers
Motivating Application

The original Chicago Ferris Wheel, built for the 1893 World's Columbian Exposition [Wikipedia]
Imagine Using a **Direct Manipulation** System...
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a Direct Manipulation System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Direct Manipulation** System…
Imagine Using a **Programmatic** System…

circle1 = ...

"Draw"
Imagine Using a **Programmatic** System…

circle1 = ...
line1 = ...
line2 = ...

"Draw"
Imagine Using a **Programmatic** System...

circle1 = ...

pt1 = left(circle1)
line1 = ...

line2 = ...

"Relate"
Imagine Using a **Programmatic** System…

```plaintext
circle1 = ...
pt1 = left(circle1)
pt2 = right(circle1)
line1 = ...
line2 = ...
```

"Relate"
Imagine Using a **Programmatic** System...

circle1 = ...

pt1 = left(circle1)
pt2 = right(circle1)
line1 = ...

pt3 = top(circle1)
pt4 = bottom(circle1)
line2 = ...

"Relate"
Imagine Using a **Programmatic** System…

```plaintext
circle1 = ...
pt1 = left(circle1)
pt2 = right(circle1)
line1 = ...

pt3 = top(circle1)
pt4 = bottom(circle1)
line2 = ...
rectArray = ...
```

"Relate"
Imagine Using a **Programmatic** System…

```javascript
function ferrisWheel(
  numSpokes,
  spokeLen,
  rotAngle,
  sizeCar,
  radiusCenter,
  cx,
  cy
) {
  ...
}

ferrisWheel(4, ...);
```
Imagine Using a **Programmatic** System...

```javascript
function ferrisWheel(
    numSpokes,
    spokeLen,
    rotAngle,
    sizeCar,
    radiusCenter,
    cx,
    cy
)
{
    ...
}

ferrisWheel(4, ...);
```

Imagine Using a **Programmatic** System…

```javascript
function ferrisWheel(
  numSpokes,
  spokeLen,
  rotAngle,
  sizeCar,
  radiusCenter,
  cx,
  cy
) {
  ...
}

ferrisWheel(8, ...);
```

"Tweak"
Imagine Using a **Programmatic** System…

```javascript
function ferrisWheel(
  numSpokes,
  spokeLen,
  rotAngle,
  sizeCar,
  radiusCenter,
  cx,
  cy
)
{
  ...
}

ferrisWheel(8, ...);
```
Imagine Using a **Programmatic** System…

```javascript
function ferrisWheel ( numSpokes , spokeLen , rotAngle , sizeCar , radiusCenter , cx , cy ) {
  ...
}

ferrisWheel(8, ...);
```

"Tweak"
Imagine Using a **Programmatic** System...

```javascript
function ferrisWheel(
  numSpokes,
  spokeLen,
  rotAngle,
  sizeCar,
  radiusCenter,
  cx,
  cy
)
{
  ...
}

ferrisWheel(8, ...);
```
Imagine Using a **Programmatic** System…

```javascript
function ferrisWheel(
  numSpokes,
  spokeLen,
  rotAngle,
  sizeCar,
  radiusCenter,
  cx,
  cy
)
{
  ...
}

ferrisWheel(16, ...);
```
Imagine Using a **Programmatic** System...

```javascript
function ferrisWheel(
    numSpokes,
    spokeLen,
    rotAngle,
    sizeCar,
    radiusCenter,
    cx,
    cy
) {
    ...
}

ferrisWheel(16, ...);
```
Imagine Using a **Programmatic + Direct Manipulation** System…

Program

Draw
Imagine Using a **Programmatic + Direct Manipulation** System…

Program

Relate
Imagine Using a **Programmatic + Direct Manipulation** System…
Imagine Using a **Programmatic + Direct Manipulation** System…
Imagine Using a **Programmatic + Direct Manipulation** System…
Imagine Using a **Programmatic** + **Direct Manipulation** System…
Imagine Using a **Programmatic + Direct Manipulation** System…
Imagine Using a **Programmatic + Direct Manipulation** System…

Programming with **Fewer Keystrokes**

Programming by **Manipulating Output**
Sketch-n-Sketch Demo

![Sketch-n-Sketch Demo Image]
Sketch-n-Sketch Demo

```
(def spokes (\(cx cy n r rot\)
  (let [rots (nPointsOnCircle n rot cx cy r)]
    (map \([x y]\) (line "lightgrey" 5 cx x y) rots))))
(defn-draw
  (spokes 183 159 10 100 0))
```
Sketch-n-Sketch Demo
def spokes(cx cy n r rot):
    (let points (PointsOnCircle n rot cx cy r))
    (map (\(x y\) (line 'lightgray 5 cx cy x y)) points))

(def newGroup4 (\(rect1.w rect1.stroke rect1.strokeWidth fill\) fill rect1.strokeWidth rect1.strokeWidth 0.5))

(def line2)
    (let [x2 y2] (+ xAnchor rect1.w (+ yAnchor rect1.h y1 x2 y2])
    (line rect1.stroke rect1.strokeWidth xAnchor yAnchor)

(def line3)
    (let [y1 x2 y2] (+ yAnchor rect1.w (+ xAnchor rect1.w y1 x2 y2])
    (line rect1.stroke rect1.strokeWidth xAnchor y2)

(def anchoredGroup (concat [ rect1 line2 line3 ]))

(def newGroup7)
    (let circle6.cx 259
        (def radius 176)

(def circle6)
    (let r 28
        (line color 469)
    (rawCircle color 360 \(circle6.cx circle6.cy r\) )

(let spokes_k4768 1Z(-1-70)
    (let spokes_k4766 -0.9528800000000001(-3.14-3.14)

(def spokes)
    (spokes circle6.cx circle6.cy spokes_k4768 radius spokes_k4766)

[anchoredGroup (concat [ ]]]
Semi-Automated Refactoring by Manipulating Output
Programming in 2015

let rect1 = ...
let line2 = ...
let line3 = ...
let rect1_x = ...
let rect1_y = ...
let rect1 = ...
let line2 = ...
let line3 = ...

Programming in 2015

Sketch-n-Sketch
let group x y size =
    let rect1 = ...
    let line2 = ...
    let line3 = ...
  group 0 0 999999
Programming in 2015

```
let group x y size =
  let rect1 = ...
  let line2 = ...
  let line3 = ...
  group 0 0 999999
```
Programming in 2015

```plaintext
let group x y size =
  let rect1 = ...
  let line2 = ...
  let line3 = ...
  group 0 0 111111
```
Draw  
Relate  
Group  
Tweak  

Programming in 2015  
Sketch-n-Sketch
(def polygon7_bot (+ (+ (* 0.5!... (def k3105 (def polygon7_top (- (* 0.5! (+... (def [polygon5_right k3038] [(-.... (def k3061 (+ polygon5_r1... (def polygon6_bot (+ (- poly... (def k3063 (+ polygon6_bo... (def polygon5_top (- polygon6_t... (def k3103 (- polygon5... (def [k3041 polygon5_bot] [(- p... (def k3134 (+ k3041 helpe... (def k3141 (+ k3038 helpe...
; Top-LevelDefs
(def rect1 ...)
(def line2 ...)
(def line3 ...)

; MainExpression
[ rect1 ... ]
Refactor
Programs of
Arbitrary
Structure
Draw

Relate

Group

Tweak

Sketch-n-Sketch

Sketch-n-Sketch ++
Draw  

Relate  

Group  

Tweak

Sketch-n-Sketch

Sketch-n-Sketch ++

Smarter Algebraic Constraint Solver
Additional Interaction to Resolve User Intent
Draw  Relate  Group  Tweak
λ-Calculus + Direct Manipulation

Reuse Effort Across Domains

Enable Experts and Library Writers to Extend Built-Ins
\(\lambda\)-Calculus + Direct Manipulation

Program

\(\lambda\)

General + Domain-Specific
\( \lambda \text{-Calculus} + \text{Direct Manipulation} \)
\(\lambda\)-Calculus + Direct Manipulation
\(\lambda\)-Calculus + Direct Manipulation
\( \lambda \)-Calculus + Direct Manipulation
λ-Calculus + Direct Manipulation
Related Work

Dynamic Drawing
Constraint-Oriented Programming
Live Programming
Programming with Fewer Keystrokes

Programming by Manipulating Output
Sketch-n-Sketch

Draw
Relate
Group
Tweak
http://ravichugh.github.io/sketch-n-sketch/
EXTRA SLIDES
Imagine Using a **Programmatic + Direct Manipulation** System…

```
[ 'svg',
  [ [ 'rect' ... ]
  [ 'circle' ... ]
]
```

```
<svg>
  <rect ... />
  <circle ... />
</svg>
```
Idea

Prototype; Refactor; Refactor; Refactor
Sketch-n-Sketch Demo

http://ravichugh.github.io/sketch-n-sketch/releases/v0.5.2/
"Raw" Shape

(x1, y1)

(x3, y3)

(x2, y2)
"Anchored" Shape

(x1, y1)

(x1+2, y1+8)

(x1+10, y1+10)
"Bounded" Shape

\[(x, y)\] = (x1 + 0.20*(x2 - x1), y1 + 0.80*(y2 - y1))
"Anchored" Group
"Anchored" Group
"Anchored" Group

"Bounded" Group
"Anchored" Group

"Bounded" Group
Which Constants to Abstract?

Heuristic: Unfrozen and Named

(def foo
  (let [a b] [1 2!]
    [a b 3]))
foo
Which Constants to Abstract?

**Heuristic: Unfrozen and Named**
Future Work

Smarter Constraint Solver for Live Sync
Reduce Dependencies on Representation
More Complex Relate Expressions
User Interactions for Ambiguities
Edit Auto-Gen Code via Direct Manipulation
END