Programming with visible data
LIVE 2018
Joshua Horowitz
data processing

data

→

data
hilbert 0 = mempty
hilbert n = hilbert' (n-1) # reflectY <> vrule 1
  <> hilbert (n-1) <> hrule 1
  <> hilbert (n-1) <> vrule (-1)
  <> hilbert' (n-1) # reflectX

  where
  hilbert' m = hilbert m # rotateBy (1/4)

diagram :: Diagram B
diagram = strokeT (hilbert 6) # lc silver
  # opacity 0.3
hilbert 0 = mempty
hilbert n = hilbert' (n-1) # reflectY <> vrule 1
    <> hilbert (n-1) <> hrule 1
    <> hilbert (n-1) <> vrule (-1)
    <> hilbert' (n-1) # reflectX

where
    hilbert' m = hilbert m # rotateBy (1/4)

diagram :: Diagram B
diagram = strokeT (hilbert 6) # lc silver
          # opacity 0.3
The Principle of Radical Visibility

All data should be visible by default.
PANE

- a prototype live programming interface
- for general data processing
- built on a functional paradigm
- in a pragmatic niche between text and VPL
- following the Principle of Radical Visibility
Demo
Related work & reflections
Max/MSP
Grasshopper
important differences
data → function → data
the data
1. PANE foregrounds the data

which provides …

• room to see the data

• handles for acting on data
2.
board to string

```
[['X',null,'O'],
 [null,'X',null],
 [null,'X','O']]
```
board to string

Input:

```
[['X', null, 'O'],
 [null, 'X', null],
 [null, 'X', 'O']]
```

Process:
1. Iterate through the input list (board).
2. For each item in the list:
   - If item is 'X', append to 'items' list.
   - If item is 'O', do nothing.
3. Map 'items' to generate rows.
4. Join rows with ' | ' separator.
5. Join all rows with '-----' separator.

Output:
```
X |   
-----
| X |
| X | O
```
board to string

input

```
[['X',null,'O'], [null,'X',null], [null,'X','O']]
```

```
<table>
<thead>
<tr>
<th>X</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>
```

```
X
```

```
X
```

```
X
```

```
+--------
<table>
<thead>
<tr>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>O</td>
</tr>
</tbody>
</table>
```

```
+--------
<table>
<thead>
<tr>
<th>X</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>O</td>
</tr>
</tbody>
</table>
```

```
+--------
<table>
<thead>
<tr>
<th>X</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>O</td>
</tr>
</tbody>
</table>
```

```
+--------
<table>
<thead>
<tr>
<th>X</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>O</td>
</tr>
</tbody>
</table>
```

```
+--------
<table>
<thead>
<tr>
<th>X</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>O</td>
</tr>
</tbody>
</table>
```

```
+--------
<table>
<thead>
<tr>
<th>X</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>O</td>
</tr>
</tbody>
</table>
```

```
X
```

```
X
```

```
X
```

```
X
```
2. PANE uses hybrid metaphors which provides ...

• legibility

• scalability
1. PANE foregrounds the data
2. PANE uses hybrid metaphors
1. Show data
2. Work with data
3. Connect with the world
1. Show data relationally (present) relationally
1. Show data relationally
2. Work with data by example
2. Work with data by example
2. Work with data by example
3. Connect with the world

Apparatus (Schachman, et al.)

an ecosystem of heterogeneous live-programming tools
Programming with visible data

LIVE 2018
Joshua Horowitz