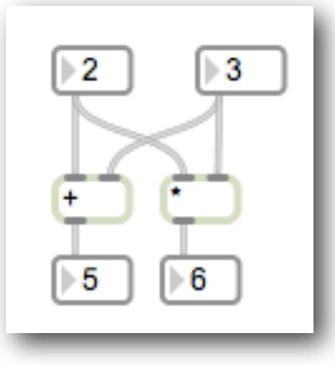
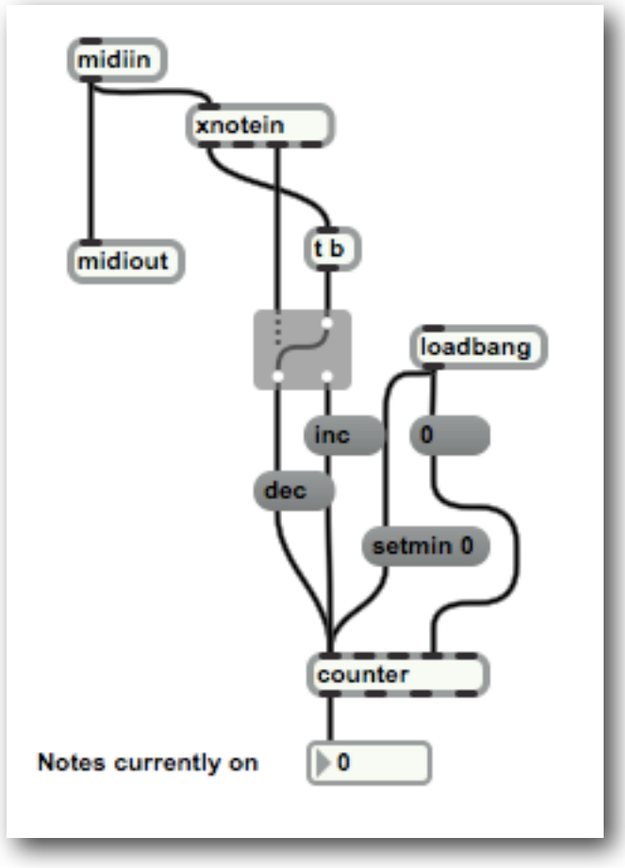
Custom UI elements for Max for Live in JavaScript

Andrew Bulhak http://dev.null.org/acb/

Max/MSP

A graphical programming environment used by artists/musicians

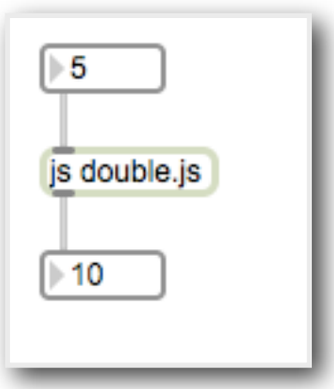




Proprietary, but related to the open-source Pd.

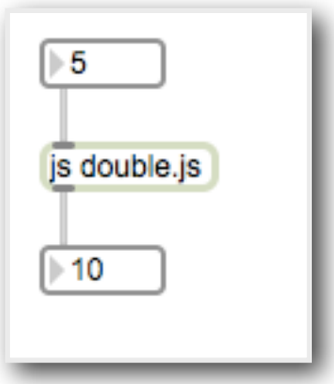
Max and JavaScript

- You can embed JavaScript code in a `js` object



The js object

- Define functions to respond to incoming data and/or messages
- Send data out with the built-in **`outlet()`** function



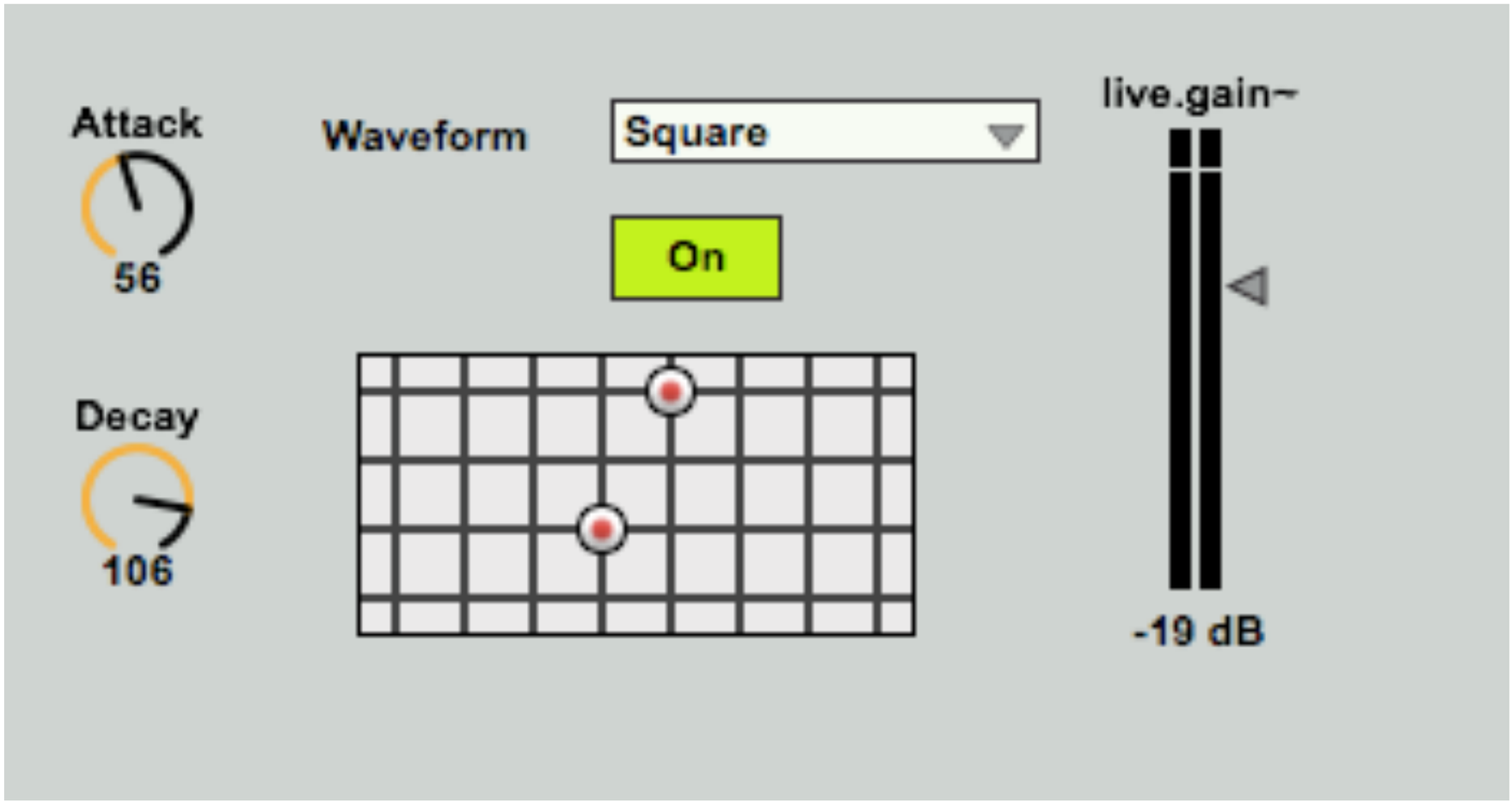

```
// double.js

function msg_int(v) {
      outlet(0, v*2);
}
```

Max for Live

Max/MSP can be used to make instruments and MIDI and audio processors for use in Ableton Live

Comes with standard control elements built-in



...though sometimes you need something different.

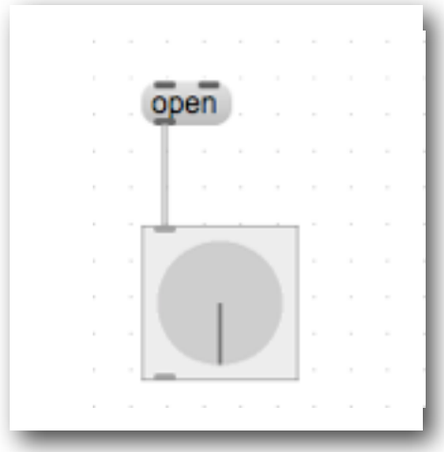
jsui

A Max node type which lets you write arbitrary graphical elements.

jsui

- Like `js`, only presents graphics and responds to mouse events
  - Takes over its rectangle of the screen
  - Define functions for handling mouse events
  - Use OpenGL-based API for drawing

- Default **`jsui`** comes with a rotary dial widget implemented in JavaScript



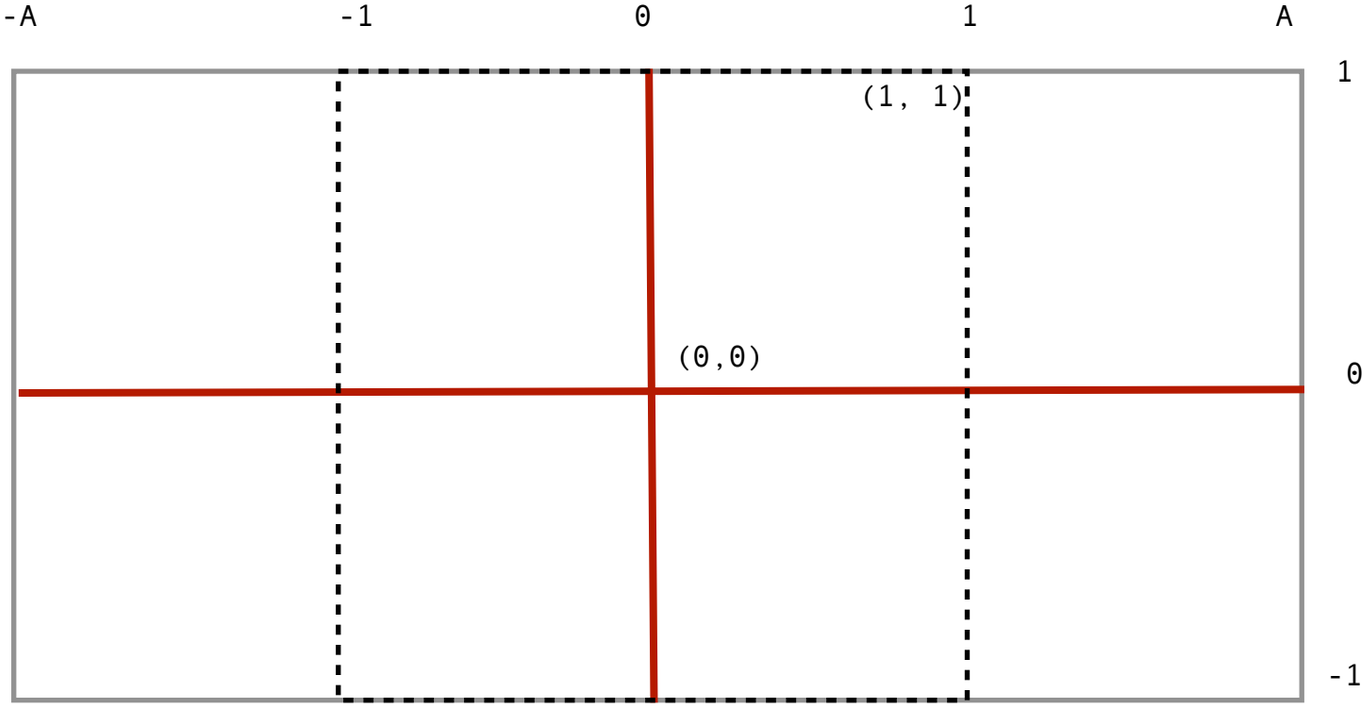
- To view/edit JavaScript source, send an **open** message

jsui gotchas

- Bears no resemblance to the HTML/CSS stack
- OpenGL-based; coordinate system is odd

What would be nice:

What jsui gives you:



This can be fixed:

...with a bit of OpenGL magic

```
with(sketch) {
    default2d();
    glmatrixmode("projection");
    glloadidentity();
    glortho(0., size[0], size[1], 0., -1,100.);
}
```

Drawing in jsui

Define a **`draw()`** method to draw your UI.
To draw, we use the built-in **`sketch`** object.

```
function draw() {
    // clear background to white
    sketch.glclearcolor(1.0, 1.0, 1.0, 1.0);
    sketch.glclear();

    // set foreground to black
    sketch.glcolor(0.0, 0.0, 0.0, 1.0);

    // draw a circle if a condition is met
    sketch.moveto(8, 8);
    if(value) {
        sketch.circle(7);
    }
};
```

Responding to clicks

Define an **`onclick()`** method to respond to mouse clicks

```
function onclick(x,y,but,cmd,shift,caps,opt,ctrl) {
    if (x < 10) {
        val = val - 1;
    } else {
        val = val + 1;
    }
    outlet(0, val);
};
```

Note that we send the value out to the patch as soon as we've changed it.

Getting values in

- We want our element to respond to incoming values, setting its internal data to them and redrawing itself.
  - Here we assume the value is an integer:

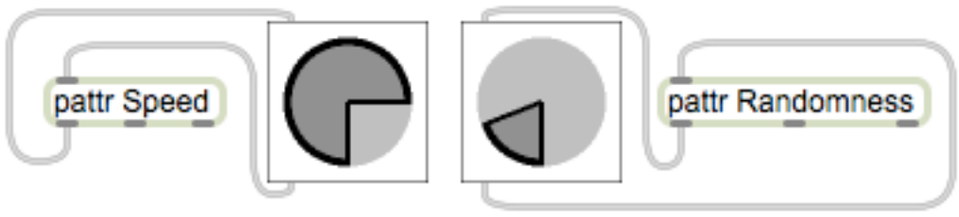
```
function msg_int(v) {
    val = v;
    draw();
    refresh();
};
```

Playing nice with Live's presets

- Ableton Live lets you save control parameters in presets and automate them.
- The built-in Live controls work with this automagically.
- Our own controls need a bit of work

pattr and parameter values

- If our control sends its value out and can receive it as input, we can make its value a Live-friendly parameter easily
- To do so, we just route its output to a **`pattr`** object, which we route into it



The **`pattr`** object must have its "Parameter mode enable" setting switched on for this to work

I have written a more detailed article on this subject, with a practical example, at tech.null.org

The article may be found at bit.ly/1aIpUze