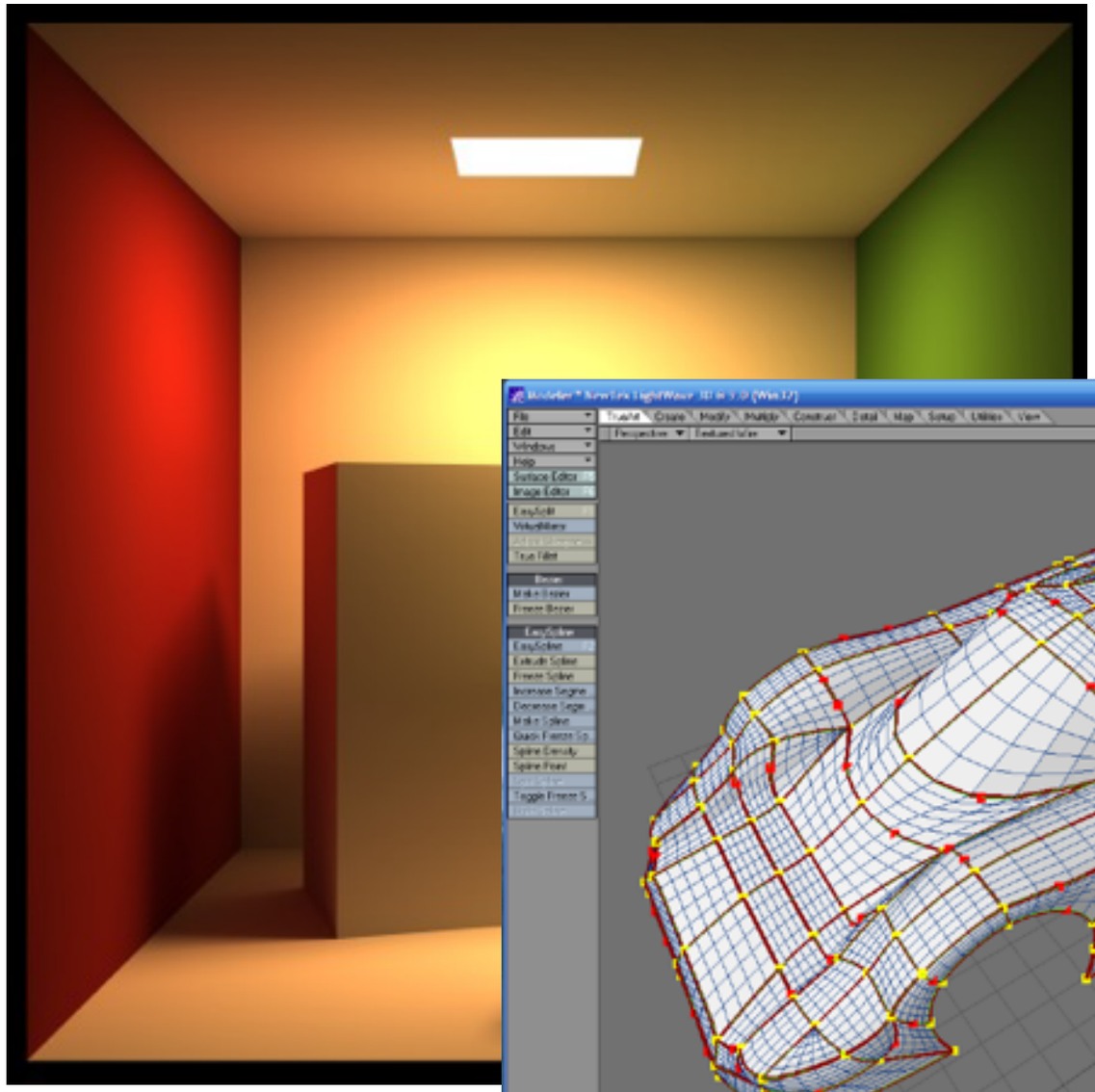
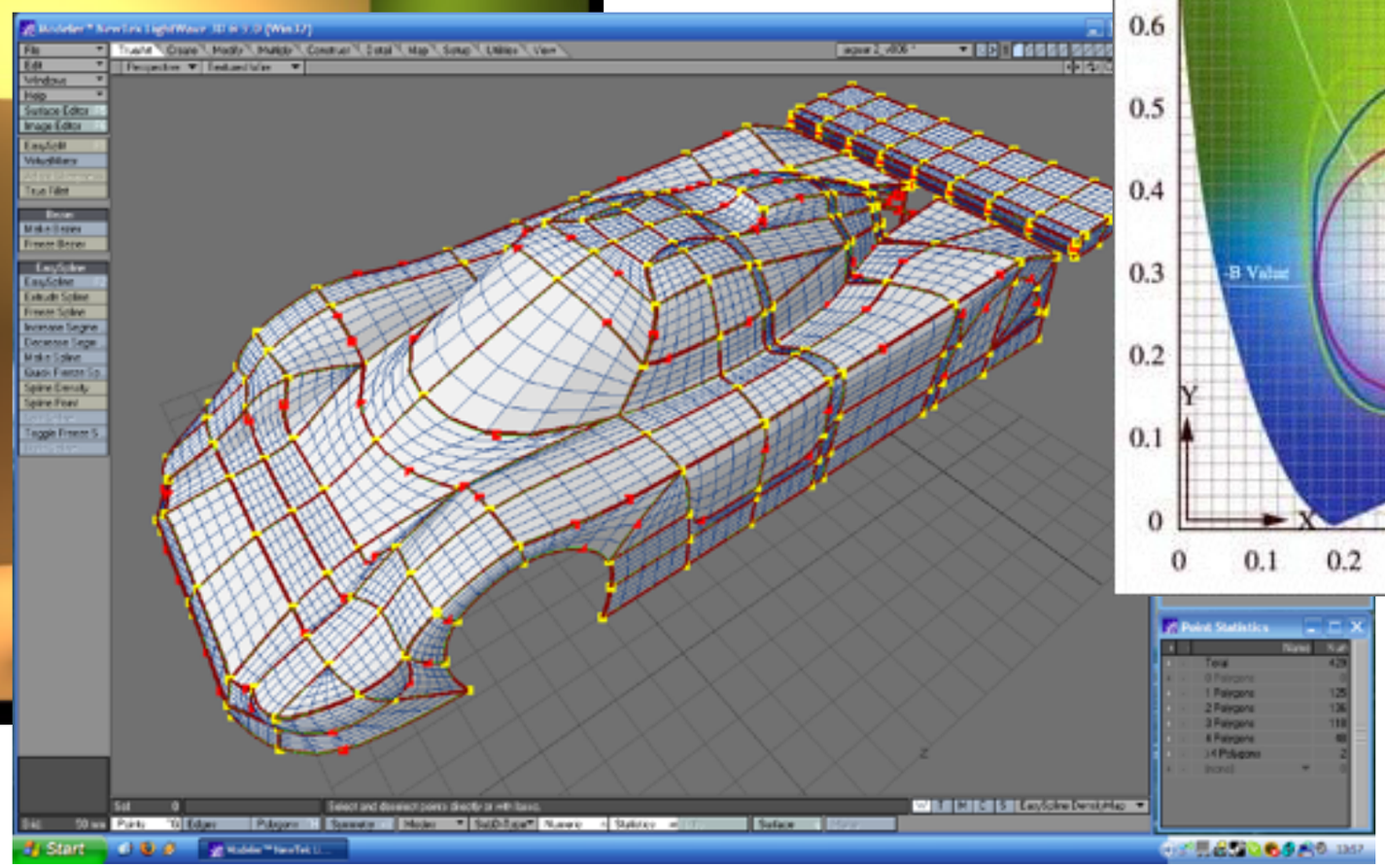
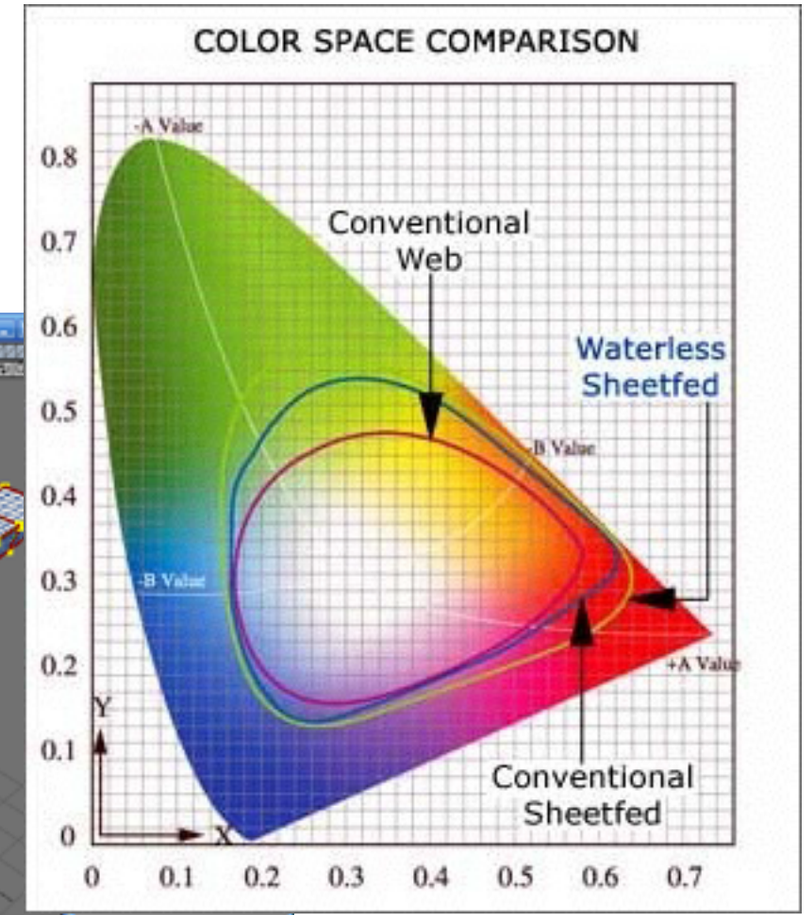






Lecture 3 : More features in Canvas 2D drawing (input elements, events/triggers, intro to transforms)

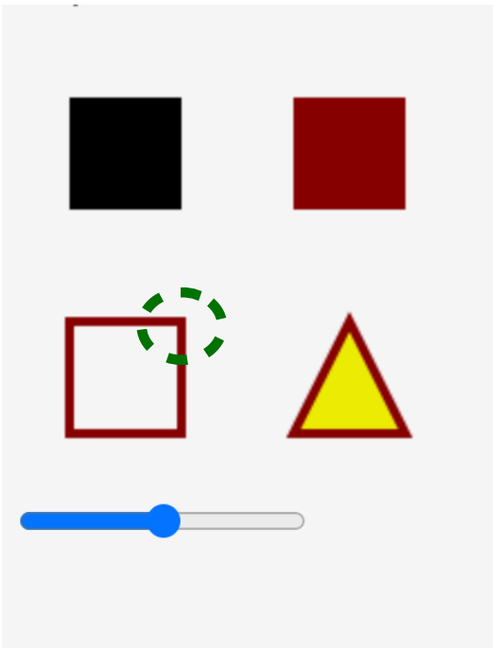
Thursday September 16th 2020

Logistics/Announcements

- We will use Canvas for posting Assignment Descriptions, rather than the Forum-like webpage that we used in past years' offering of 559 (alternatives were explored, not looking all that sustainable)
- We will still leverage some pointers to tutorials that have been put together in the past; those will be linked from your Assignments, and they do work.
- We'll continue to use Piazza as the main communication hub.

Your first programming assignment

- Where do I find it?
  - Look at the Assignment Description page on Canvas (link) which is also where you will turn in your work.
  - TL;DR: Create a simple 2D drawing, as an html page (not in JSbin) and include both non-filled strokes and filled shapes. Use slider(s) to change something in the drawing. If submitting more than a single .html, put in a .zip.



Maybe moving the slider changes the location of this vertex …

Your first programming assignment

- Deadline is Friday Sept 24th @ midnight (late turn-ins till 9/29; check late policy on assignment description)
  - The description of the programming assignment includes links to reference materials, and a few brief tutorials. *Please review these materials! You will find them handy!*
  - You are given a link to a public GitHub repository where the demos (shown in class via JSbin) will be duplicated, but in the form that you are expected to develop your own programs. Study those, and run them in browser.
  - Your hand-in cannot be a JSbin link. You need to provide an actual .html file (ideally with a separated-out .js file for the JavaScript code). We'll see examples of this.

github.com/sifakis/CS559F21_Demos

Apps

Search or jump to... / Pull requests Issues Marketplace Explore

sifakis / CS559F21_Demos Public

Un

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

Eftychios Sifakis Initial commit (Week2)

Week2	Initial commit (Week2)
LICENSE	Initial commit (Week2)
README.md	Initial commit (Week2)

Clone

HTTPS SSH GitHub CLI

https://github.com/sifakis/CS559F21_De

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

README.md

CS559F21_Demos

Software artifacts and Demos for CS559 (Fall 2021) "Computer Graphics"

github.com/sifakis/CS559F21_Demos

Apps

Search or jump to...

This is a public repository (you can download/clone it, but not write to it; anyone can, in fact)

sifakis / **CS559F21_Demos** Public

Un

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

main | 1 branch | 0 tags

Go to file | Add file | Code

Eftychios Sifakis Initial commit (Week2)

Week2	Initial commit (Week2)
LICENSE	Initial commit (Week2)
README.md	Initial commit (Week2)

Clone

HTTPS SSH GitHub CLI

`https://github.com/sifakis/CS559F21_De`

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

README.md

CS559F21_Demos

Software artifacts and Demos for CS559 (Fall 2021) "Computer Graphics"

github.com/sifakis/CS559F21_Demos

Apps

Search or jump to...

You might be familiar with the Git version control system … We'll go through practical use "by example" in a limited capacity, but there are many good tutorials (e.g. this)

sifakis / CS559F21_Demos Public

Un

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

main | 1 branch | 0 tags | Go to file | Add file | Code

Eftychios Sifakis Initial commit (Week2)

Week2	Initial commit (Week2)
LICENSE	Initial commit (Week2)
README.md	Initial commit (Week2)

Clone

HTTPS SSH GitHub CLI

`https://github.com/sifakis/CS559F21_De`

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

README.md

CS559F21_Demos

Software artifacts and Demos for CS559 (Fall 2021) "Computer Graphics"

github.com/sifakis/CS559F21_Demos

Apps

Search or jump to... / Pull requests Issues Marketplace Explore

sifakis / CS559F21_Demos Public

Un

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

Eftychios Sifakis Initial commit (Week2)

Week2	Initial commit (Week2)
LICENSE	Initial commit (Week2)
README.md	Initial commit (Week2)

Clone

HTTPS SSH GitHub CLI

https://github.com/sifakis/CS559F21_De

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

README.md

CS559F21_Demos

Quickest way to start … download a .zip archive

Software artifacts and Demos for CS559 (Fall 2021) "Computer Graphics"

github.com/sifakis/CS559F21_Demos

Apps

Search or jump to... / Pull requests Issues Marketplace Explore

sifakis / CS559F21_Demos Public

Un

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

Eftychios Sifakis Initial commit (Week2)

Week2	Initial commit (Week2)
LICENSE	Initial commit (Week2)
README.md	Initial commit (Week2)

Clone

HTTPS SSH GitHub CLI

https://github.com/sifakis/CS559F21_De

Use Git or checkout with SVN using the web URL.

*Even better … **clone** the repository to work locally (check out the tutorial)*

README.md

CS559F21_Demos

Software artifacts and Demos for CS559 (Fall 2021) "Computer Graphics"

github.com/sifakis/CS559F21_Demos

Apps

Search or jump to... /

Pull requests Issues Marketplace Explore

sifakis / CS559F21_Demos Public

Un

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

Demo0 — -bash — 95×20

```
eftychioss-mini:Development sifakis$ git clone https://github.com/sifakis/CS559F21_Demos.git
Cloning into 'CS559F21_Demos'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 18 (delta 3), reused 18 (delta 3), pack-reused 0
Unpacking objects: 100% (18/18), done.
eftychioss-mini:Development sifakis$ cd CS559F21_Demos/
eftychioss-mini:CS559F21_Demos sifakis$ ls
LICENSE         README.md       Week2
eftychioss-mini:CS559F21_Demos sifakis$ cd Week2/
eftychioss-mini:Week2 sifakis$ ls
Demo0   Demo1   Demo2   Demo3
eftychioss-mini:Week2 sifakis$ cd Demo0/
eftychioss-mini:Demo0 sifakis$ ls
demo.html       demo.js
eftychioss-mini:Demo0 sifakis$
```

Then, simply "open" the .html file with your browser …

Try right-click and then …
… "View Page Source" to get to the HTML source (browser dependent)
… click on "demo.js" to get the JavaScript source (here it's separate)
… or "Inspect" to launch the debugger (in Chrome)

The debugger within Chrome (similar in other browsers)

File | /Users/sifakis/Development/CS559F20_Demos/Week2/Demo0/demo.html

Apps

Elements Console Sources Network Performance

Page Filesystem

top
file://
Users/sifakis/Development/CS
demo.html
demo.js

demo.js

```
1  var canvas = document.getElementById('myCanvas');
2  var context = canvas.getContext('2d');
3
4  // a thick red line
5  context.lineWidth = 5;
6  context.strokeStyle = "red";
7
8  // the actual line
9  context.beginPath();
10 context.moveTo(50,50);
11 context.lineTo(100,100);
12 context.stroke();
13
```

{} Line 9, Column 1 Coverage: n/a

Call Stack

Not paused

Breakpoints

demo.js:9
context.beginPath();

Scope Watch

context.lineWidth: 5

The debugger within Chrome (similar in other browsers)

File | /Users/sifakis/Development/CS559F20_Demos/Week2/Demo0/demo.html

Apps

Elements | Console | Sources | Network | Performance

Page | Filesystem

top
file://
Users/sifakis/Development/CS
demo.html
demo.js

demo.js

```
1  var canvas = document.getElementById('myCanvas');
2  var context = canvas.getContext('2d');
3
4  // a thick red line
5  context.lineWidth = 5;
6  context.strokeStyle = "red";
7
8  // the actual line
9  context.beginPath();
10 context.moveTo(50,50);
11 context.lineTo(100,100);
12 context.stroke();
13
```

{} Line 9, Column 1 Coverage: n/a

Call Stack

Not paused

Breakpoints

demo.js:9
context.beginPath();

Scope | Watch

context.lineWidth: 5

Many interface elements …

What do the individual files (.html, js) look like?

Week2/Demo0

demo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>A simple drawing demo</title>
  </head>
  <body>
    <canvas id="myCanvas"
       width="200" height="200"
       style="border:1px solid #000;">
    </canvas>
    <script src="demo.js" id="module"></script>
  </body>
</html>
```

demo.js

```
var canvas = document.getElementById('myCanvas');
var context = canvas.getContext('2d');

// a thick red line
context.lineWidth = 5;
context.strokeStyle = "red";

// the actual line
context.beginPath();
context.moveTo(50,50);
context.lineTo(100,100);
context.stroke();
```

Difference from JSbin (or JSbin-downloaded monolithic html file) :
The JavaScript file is separate, and is specified as the source of the script

Is this the only way?

- Running your code in your browser, from a local copy, is infinitely better than via JSbin (for reasons discussed)
- Even better: Host the page on a web-server
  - This allows much broader functionality, as browsers are peculiar with how they allow manipulation of files if not run over a web server
  - Instead of hosting the page on the web (your homepage?) you can run it via a local server
  - We may talk about this option in greater detail, later (if the nature of assignments strongly suggests it)

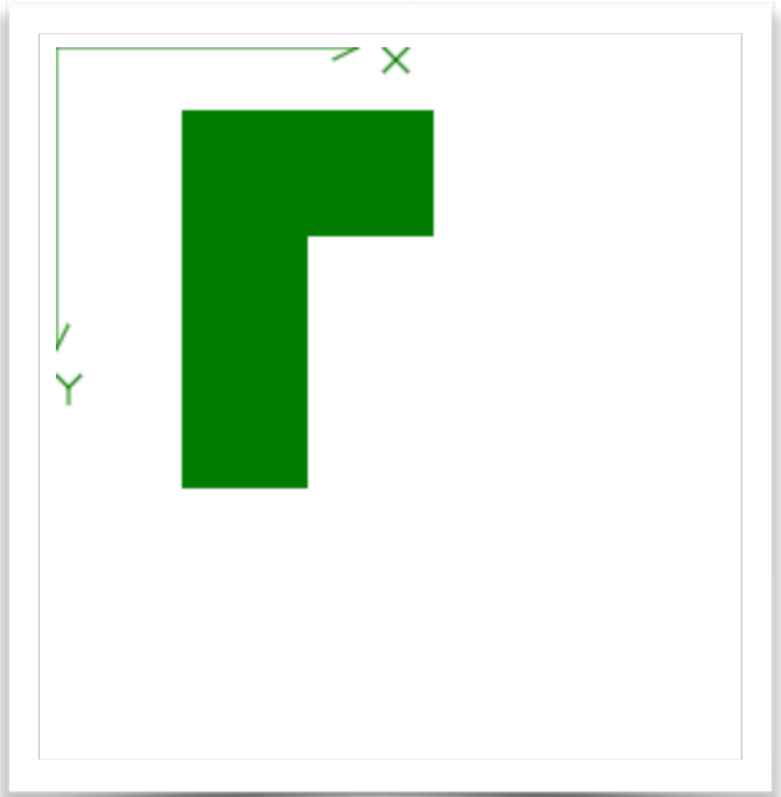
Additional features (JavaScript/Canvas)

jsbin.com/suhujar

Week2/Demo1

demo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Coordinate axes and a filled shape</title>
  </head>
  <body>
    <canvas id="myCanvas"
        width="400" height="400">
    </canvas>
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



demo.js

```
var canvas = document.getElementById('myCanvas');

function draw() {
  var context = canvas.getContext('2d');

  function DrawLshape(color) {
    context.beginPath();
    context.fillStyle = color;
    context.moveTo(50,25);
    context.lineTo(150,25);
    context.lineTo(150,75);
    context.lineTo(100,75);
    context.lineTo(100,175);
    context.lineTo(50,175);
    context.closePath();
    context.fill();
  }

  function DrawAxes(color) {
    context.strokeStyle=color;
    context.beginPath();
    // Axes
    context.moveTo(120,0);context.lineTo(0,0);context.lineTo(0,120);
    // Arrowheads
    context.moveTo(110,5);context.lineTo(120,0);context.lineTo(110,-5);
    context.moveTo(5,110);context.lineTo(0,120);context.lineTo(-5,110);
    // X-label
    context.moveTo(130,0);context.lineTo(140,10);
    context.moveTo(130,10);context.lineTo(140,0);
    // Y-label
    context.moveTo(0,130);context.lineTo(5,135);context.lineTo(10,130);
    context.moveTo(5,135);context.lineTo(5,142);

    context.stroke();
   }

  // make sure you understand these
  DrawAxes("green");
  DrawLshape("green");

}
draw();
```

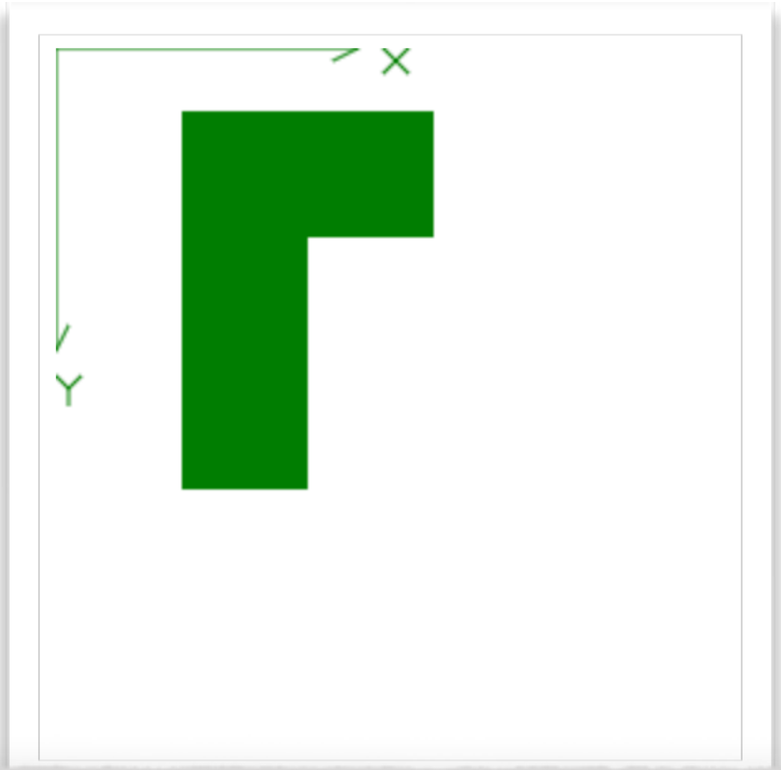
Additional features (JavaScript/Canvas)

jsbin.com/suhujar

Week2/Demo1

demo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Coordinate axes and a filled shape</title>
  </head>
  <body>
    <canvas id="myCanvas"
        width="400" height="400">
    </canvas>
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



Functions, nested functions, and calls (with parameters)

demo.js

```
var canvas = document.getElementById('myCanvas');

function draw() {
  var context = canvas.getContext('2d');

  function DrawLshape(color) {
    context.beginPath();
    context.fillStyle = color;
    context.moveTo(50,25);
    context.lineTo(150,25);
    context.lineTo(150,75);
    context.lineTo(100,75);
    context.lineTo(100,175);
    context.lineTo(50,175);
    context.closePath();
    context.fill();
  }

  function DrawAxes(color) {
    context.strokeStyle=color;
    context.beginPath();
    // Axes
    context.moveTo(120,0);context.lineTo(0,0);context.lineTo(0,120);
    // Arrowheads
    context.moveTo(110,5);context.lineTo(120,0);context.lineTo(110,-5);
    context.moveTo(5,110);context.lineTo(0,120);context.lineTo(-5,110);
    // X-label
    context.moveTo(130,0);context.lineTo(140,10);
    context.moveTo(130,10);context.lineTo(140,0);
    // Y-label
    context.moveTo(0,130);context.lineTo(5,135);context.lineTo(10,130);
    context.moveTo(5,135);context.lineTo(5,142);

    context.stroke();
   }

  // make sure you understand these
  DrawAxes("green");
  DrawLshape("green");

}
draw();
```

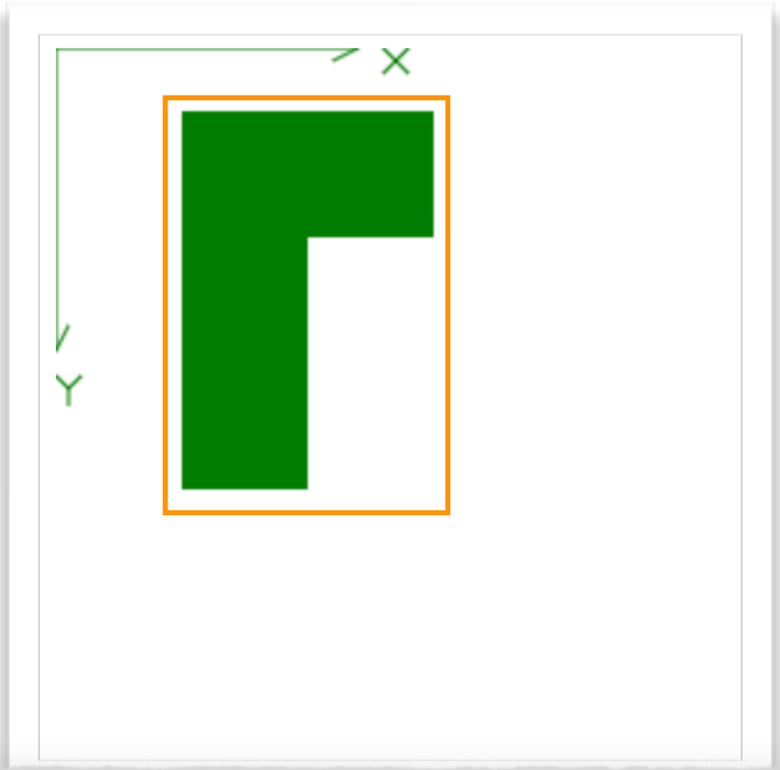
Additional features (JavaScript/Canvas)

jsbin.com/suhujar

Week2/Demo1

demo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Coordinate axes and a filled shape</title>
  </head>
  <body>
    <canvas id="myCanvas"
        width="400" height="400">
    </canvas>
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



Drawing closed, filled polygons, with specified fillStyle (vs strokeStyle)

demo.js

```
var canvas = document.getElementById('myCanvas');

function draw() {
  var context = canvas.getContext('2d');

  function DrawLshape(color) {
    context.beginPath();
    context.fillStyle = color;
    context.moveTo(50,25);
    context.lineTo(150,25);
    context.lineTo(150,75);
    context.lineTo(100,75);
    context.lineTo(100,175);
    context.lineTo(50,175);
    context.closePath();
    context.fill();
  }

  function DrawAxes(color) {
    context.strokeStyle=color;
    context.beginPath();
    // Axes
    context.moveTo(120,0);context.lineTo(0,0);context.lineTo(0,120);
    // Arrowheads
    context.moveTo(110,5);context.lineTo(120,0);context.lineTo(110,-5);
    context.moveTo(5,110);context.lineTo(0,120);context.lineTo(-5,110);
    // X-label
    context.moveTo(130,0);context.lineTo(140,10);
    context.moveTo(130,10);context.lineTo(140,0);
    // Y-label
    context.moveTo(0,130);context.lineTo(5,135);context.lineTo(10,130);
    context.moveTo(5,135);context.lineTo(5,142);

    context.stroke();
   }

  // make sure you understand these
  DrawAxes("green");
  DrawLshape("green");

}
draw();
```

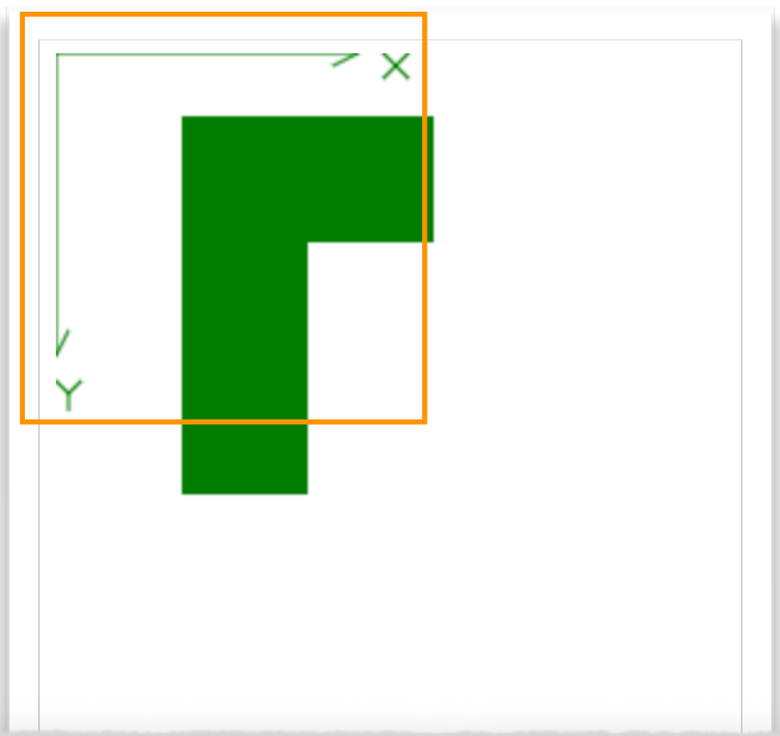
Additional features (JavaScript/Canvas)

jsbin.com/suhujar

Week2/Demo1

demo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Coordinate axes and a filled shape</title>
  </head>
  <body>
    <canvas id="myCanvas"
        width="400" height="400">
    </canvas>
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



Drawing several (disconnected) chains of line segments (notice moveTo/lineTo succession)

demo.js

```
var canvas = document.getElementById('myCanvas');

function draw() {
  var context = canvas.getContext('2d');

  function DrawLshape(color) {
    context.beginPath();
    context.fillStyle = color;
    context.moveTo(50,25);
    context.lineTo(150,25);
    context.lineTo(150,75);
    context.lineTo(100,75);
    context.lineTo(100,175);
    context.lineTo(50,175);
    context.closePath();
    context.fill();
  }

  function DrawAxes(color) {
    context.strokeStyle=color;
    context.beginPath();
    // Axes
    context.moveTo(120,0);context.lineTo(0,0);context.lineTo(0,120);
    // Arrowheads
    context.moveTo(110,5);context.lineTo(120,0);context.lineTo(110,-5);
    context.moveTo(5,110);context.lineTo(0,120);context.lineTo(-5,110);
    // X-label
    context.moveTo(130,0);context.lineTo(140,10);
    context.moveTo(130,10);context.lineTo(140,0);
    // Y-label
    context.moveTo(0,130);context.lineTo(5,135);context.lineTo(10,130);
    context.moveTo(5,135);context.lineTo(5,142);

    context.stroke();
  }

  // make sure you understand these
  DrawAxes("green");
  DrawLshape("green");

}
draw();
```

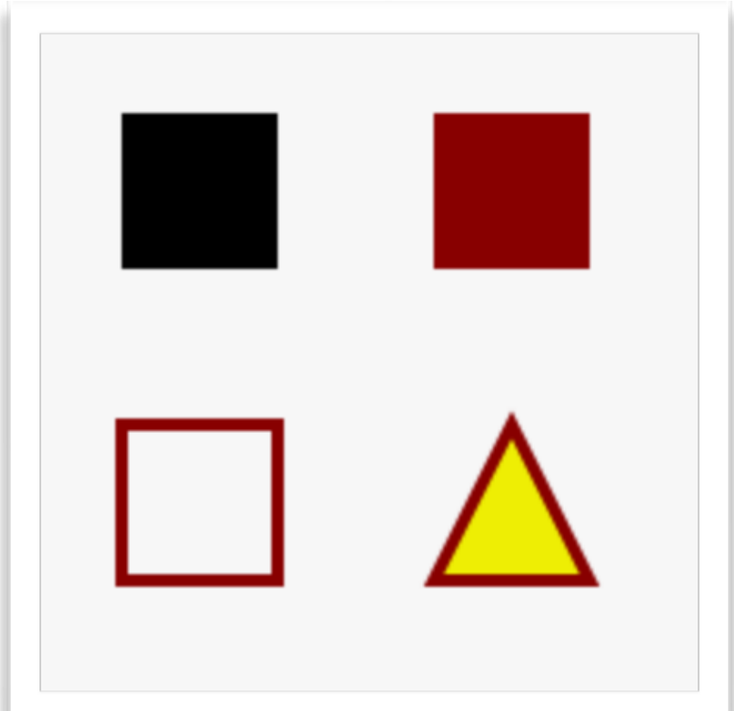
Additional features (JavaScript/Canvas)

jsbin.com/kisuhox

Week2/Demo3

demo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>A few static shapes</title>
  </head>
  <body>
    <canvas id="myCanvas"
       width="200" height="200">
    </canvas>
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



demo.js

```
// function draw() {
window.onload = function() {
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');
    context.beginPath();
    context.rect(25,25,50,50);
    context.fill();

    context.beginPath();
    context.rect(125,25,50,50);
    context.fillStyle = "#800";
    context.fill();

    context.beginPath();
    context.rect(25,125,50,50);
    context.strokeStyle = "#800";
    context.lineWidth = 4;
    context.stroke();

    context.beginPath();
    context.moveTo(150,125);
    context.lineTo(125,175);
    context.lineTo(175,175);
    context.closePath();
    context.fillStyle="#EE0";
    context.fill();
    context.stroke();
};
// window.onload = draw;
```

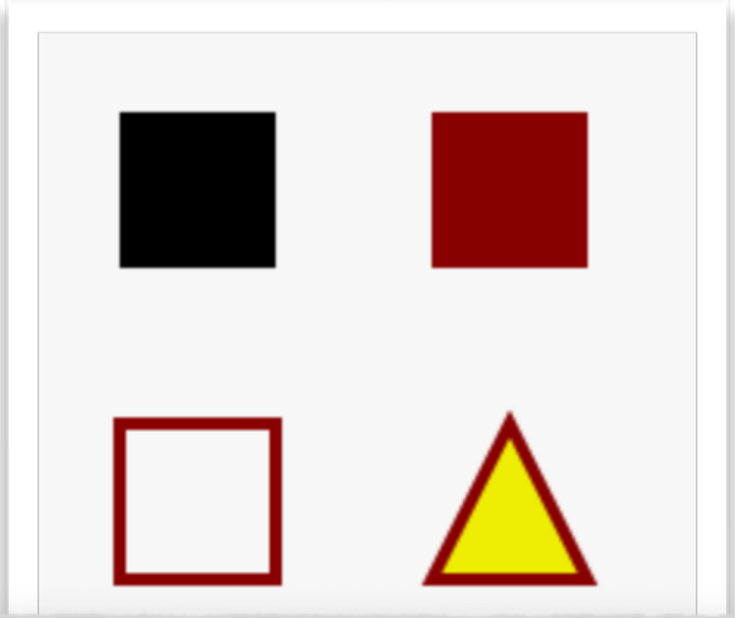
Additional features (JavaScript/Canvas)

jsbin.com/kisuhox

Week2/Demo3

demo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>A few static shapes</title>
  </head>
  <body>
    <canvas id="myCanvas"
       width="200" height="200">
    </canvas>
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



demo.js

```
// function draw() {
window.onload = function() {
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');
    context.beginPath();
    context.rect(25,25,50,50);
    context.fill();

    context.beginPath();
    context.rect(125,25,50,50);
    context.fillStyle = "#800";
    context.fill();

    context.beginPath();
    context.rect(25,125,50,50);
    context.strokeStyle = "#800";
    context.lineWidth = 4;
    context.stroke();

    context.beginPath();
    context.moveTo(150,125);
    context.lineTo(125,175);
    context.lineTo(175,175);
    context.closePath();
    context.fillStyle="#EE0";
    context.fill();
    context.stroke();
};
// window.onload = draw;
```

Specifying which function is run upon "loading" of the page (check tutorial)
Also, contrast 2 forms of syntax …

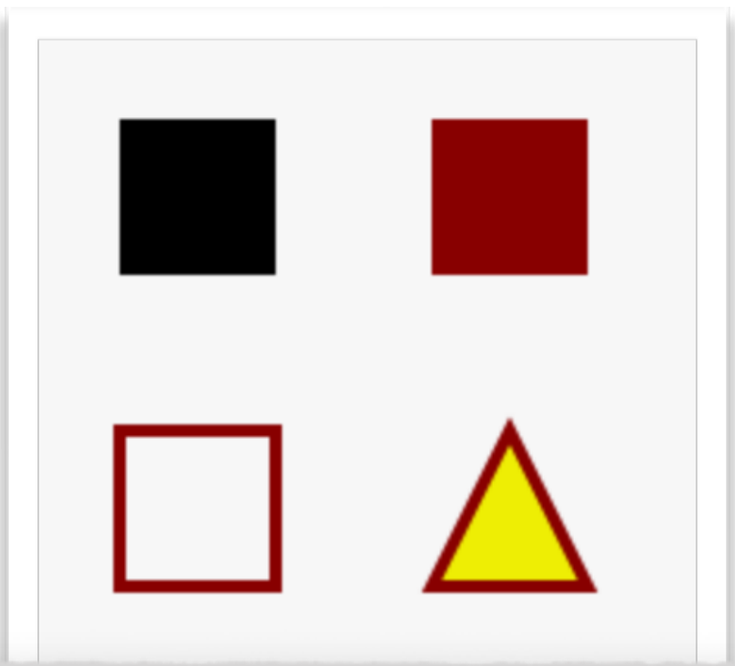
Additional features (JavaScript/Canvas)

jsbin.com/kisuhox

Week2/Demo3

demo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>A few static shapes</title>
  </head>
  <body>
    <canvas id="myCanvas"
       width="200" height="200">
    </canvas>
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



Drawing rectangles, and drawing polygons with different fill color and different outline color

demo.js

```
// function draw() {
window.onload = function() {
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');
    context.beginPath();
    context.rect(25,25,50,50);
    context.fill();

    context.beginPath();
    context.rect(125,25,50,50);
    context.fillStyle = "#800";
    context.fill();

    context.beginPath();
    context.rect(25,125,50,50);
    context.strokeStyle = "#800";
    context.lineWidth = 4;
    context.stroke();

    context.beginPath();
    context.moveTo(150,125);
    context.lineTo(125,175);
    context.lineTo(175,175);
    context.closePath();
    context.fillStyle="#EE0";
    context.fill();
    context.stroke();
};
// window.onload = draw;
```

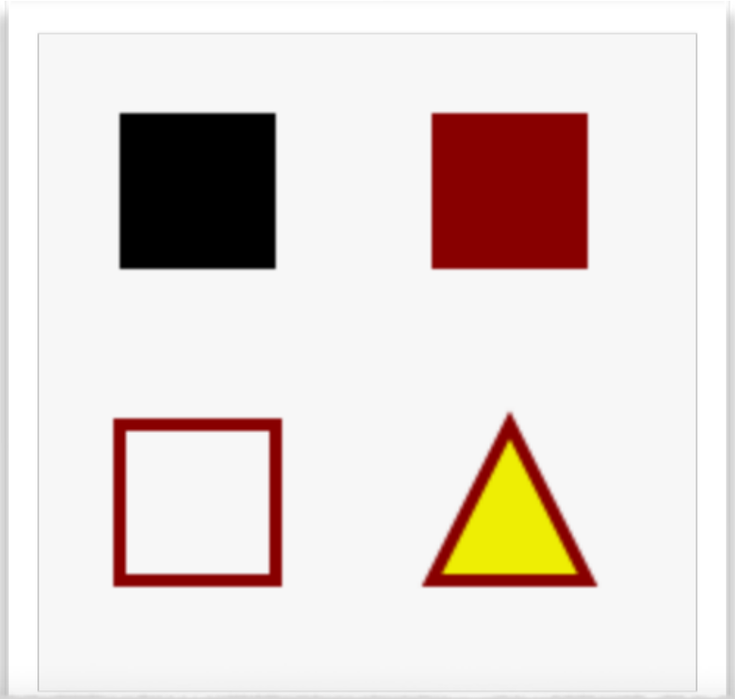
Additional features (JavaScript/Canvas)

jsbin.com/kisuhox

Week2/Demo3

demo.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>A few static shapes</title>
  </head>
  <body>
    <canvas id="myCanvas"
       width="200" height="200">
    </canvas>
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



Note alternate specification for color (see this tutorial as well)

demo.js

```javascript
// function draw() {
window.onload = function() {
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');
    context.beginPath();
    context.rect(25,25,50,50);
    context.fill();

    context.beginPath();
    context.rect(125,25,50,50);
    context.fillStyle = "#800";
    context.fill();

    context.beginPath();
    context.rect(25,125,50,50);
    context.strokeStyle = "#800";
    context.lineWidth = 4;
    context.stroke();

    context.beginPath();
    context.moveTo(150,125);
    context.lineTo(125,175);
    context.lineTo(175,175);
    context.closePath();
    context.fillStyle="#EE0";
    context.fill();
    context.stroke();
};
// window.onload = draw;
```

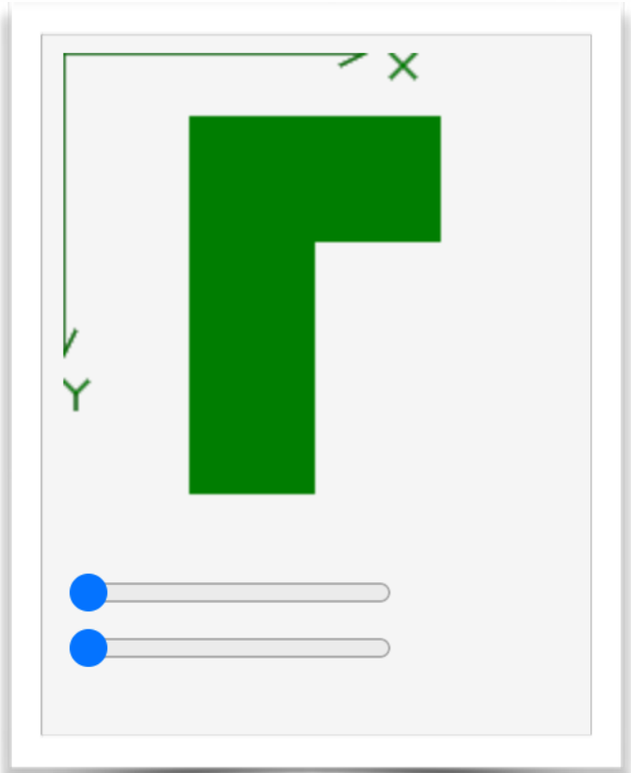
Additional features (JavaScrip

jsbin.com/fesukexori

Week2/Demo2

demo.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Simple demonstration of slider interface</title>
  </head>
  <body>
    <canvas id="myCanvas"
        width="400" height="400">
    </canvas>
    <br/>
    <input id="slider1" type="range" min="0" max="100" />
    <br/>
    <input id="slider2" type="range" min="0" max="100" />
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



demo.js

```javascript
function setup() {
  var canvas = document.getElementById('myCanvas');
  var slider1 = document.getElementById('slider1');
  slider1.value = 0;
  var slider2 = document.getElementById('slider2');
  slider2.value = 0;
  function draw() {
    var context = canvas.getContext('2d');
    canvas.width = canvas.width;
    // use the sliders to get various parameters
    var dx = slider1.value;
    var dy = slider2.value;

    function DrawLshape(color) {
      context.beginPath();
      context.fillStyle = color;
      context.moveTo(50,25);context.lineTo(150,25);context.lineTo(150,75);
      context.lineTo(100,75);context.lineTo(100,175);context.lineTo(50,175);
      context.closePath();
      context.fill();
    }

    function DrawAxes(color) {
      context.strokeStyle=color;
      context.beginPath();
      // Axes
      context.moveTo(120,0);context.lineTo(0,0);context.lineTo(0,120);
      // Arrowheads
      context.moveTo(110,5);context.lineTo(120,0);context.lineTo(110,-5);
      context.moveTo(5,110);context.lineTo(0,120);context.lineTo(-5,110);
      // X-label
      context.moveTo(130,0);context.lineTo(140,10);
      context.moveTo(130,10);context.lineTo(140,0);
      // Y-label
      context.moveTo(0,130);context.lineTo(5,135);context.lineTo(10,130);
      context.moveTo(5,135);context.lineTo(5,142);

      context.stroke();
     }

    // make sure you understand these

    DrawAxes("black");
    context.save();
    context.translate(dx,dy);
    DrawAxes("green");
    DrawLshape("green");
    context.restore();
  }
  slider1.addEventListener("input",draw);
  slider2.addEventListener("input",draw);
  draw();
}
window.onload = setup;
```

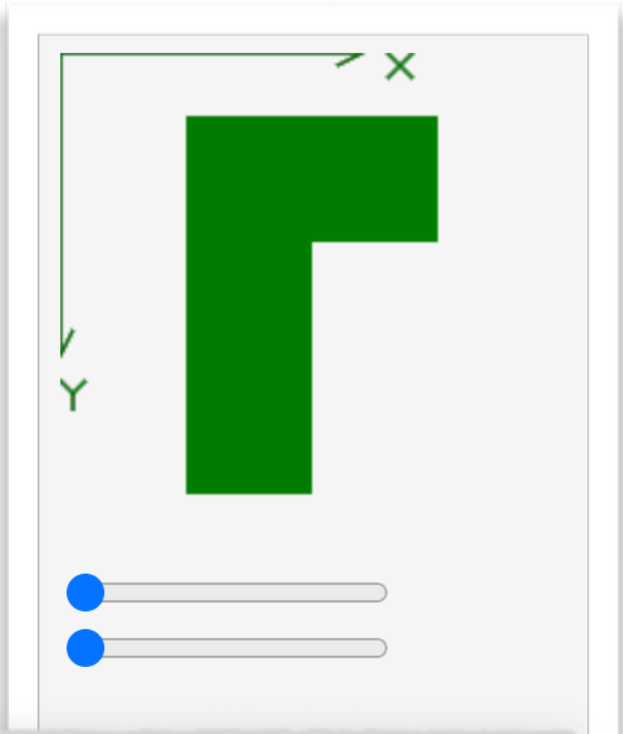
Additional features (JavaScrip

jsbin.com/fesukexori

Week2/Demo2

demo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple demonstration of slider interface</title>
  </head>
  <body>
    <canvas id="myCanvas"
        width="400" height="400">
    </canvas>
    <br/>
    <input id="slider1" type="range" min="0" max="100" />
    <br/>
    <input id="slider2" type="range" min="0" max="100" />
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



Sliders (declaration, initialization, retrieval of values)

demo.js

```
function setup() {
  var canvas = document.getElementById('myCanvas');
  var slider1 = document.getElementById('slider1');
  slider1.value = 0;
  var slider2 = document.getElementById('slider2');
  slider2.value = 0;
  function draw() {
    var context = canvas.getContext('2d');
    canvas.width = canvas.width;
    // use the sliders to get various parameters
    var dx = slider1.value;
    var dy = slider2.value;

    function DrawLshape(color) {
      context.beginPath();
      context.fillStyle = color;
      context.moveTo(50,25);context.lineTo(150,25);context.lineTo(150,75);
      context.lineTo(100,75);context.lineTo(100,175);context.lineTo(50,175);
      context.closePath();
      context.fill();
    }

    function DrawAxes(color) {
      context.strokeStyle=color;
      context.beginPath();
      // Axes
      context.moveTo(120,0);context.lineTo(0,0);context.lineTo(0,120);
      // Arrowheads
      context.moveTo(110,5);context.lineTo(120,0);context.lineTo(110,-5);
      context.moveTo(5,110);context.lineTo(0,120);context.lineTo(-5,110);
      // X-label
      context.moveTo(130,0);context.lineTo(140,10);
      context.moveTo(130,10);context.lineTo(140,0);
      // Y-label
      context.moveTo(0,130);context.lineTo(5,135);context.lineTo(10,130);
      context.moveTo(5,135);context.lineTo(5,142);

      context.stroke();
     }

    // make sure you understand these

    DrawAxes("black");
    context.save();
    context.translate(dx,dy);
    DrawAxes("green");
    DrawLshape("green");
    context.restore();
  }
  slider1.addEventListener("input",draw);
  slider2.addEventListener("input",draw);
  draw();
}
window.onload = setup;
```

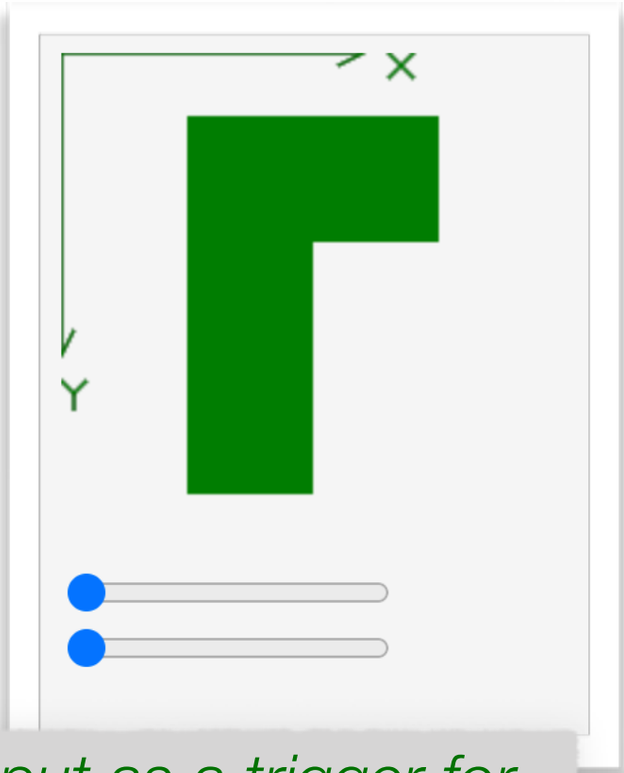
Additional features (JavaScrip

jsbin.com/fesukexori

Week2/Demo2

demo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple demonstration of slider interface</title>
  </head>
  <body>
    <canvas id="myCanvas"
        width="400" height="400">
    </canvas>
    <br/>
    <input id="slider1" type="range" min="0" max="100" />
    <br/>
    <input id="slider2" type="range" min="0" max="100" />
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



Using slider input as a trigger for re-drawing (again, tutorial)

demo.js

```
function setup() {
  var canvas = document.getElementById('myCanvas');
  var slider1 = document.getElementById('slider1');
  slider1.value = 0;
  var slider2 = document.getElementById('slider2');
  slider2.value = 0;
  function draw() {
    var context = canvas.getContext('2d');
    canvas.width = canvas.width;
    // use the sliders to get various parameters
    var dx = slider1.value;
    var dy = slider2.value;

    function DrawLshape(color) {
      context.beginPath();
      context.fillStyle = color;
      context.moveTo(50,25);context.lineTo(150,25);context.lineTo(150,75);
      context.lineTo(100,75);context.lineTo(100,175);context.lineTo(50,175);
      context.closePath();
      context.fill();
    }

    function DrawAxes(color) {
      context.strokeStyle=color;
      context.beginPath();
      // Axes
      context.moveTo(120,0);context.lineTo(0,0);context.lineTo(0,120);
      // Arrowheads
      context.moveTo(110,5);context.lineTo(120,0);context.lineTo(110,-5);
      context.moveTo(5,110);context.lineTo(0,120);context.lineTo(-5,110);
      // X-label
      context.moveTo(130,0);context.lineTo(140,10);
      context.moveTo(130,10);context.lineTo(140,0);
      // Y-label
      context.moveTo(0,130);context.lineTo(5,135);context.lineTo(10,130);
      context.moveTo(5,135);context.lineTo(5,142);

      context.stroke();
     }

    // make sure you understand these

    DrawAxes("black");
    context.save();
    context.translate(dx,dy);
    DrawAxes("green");
    DrawLshape("green");
    context.restore();
  }
  slider1.addEventListener("input",draw);
  slider2.addEventListener("input",draw);
  draw();
}
window.onload = setup;
```

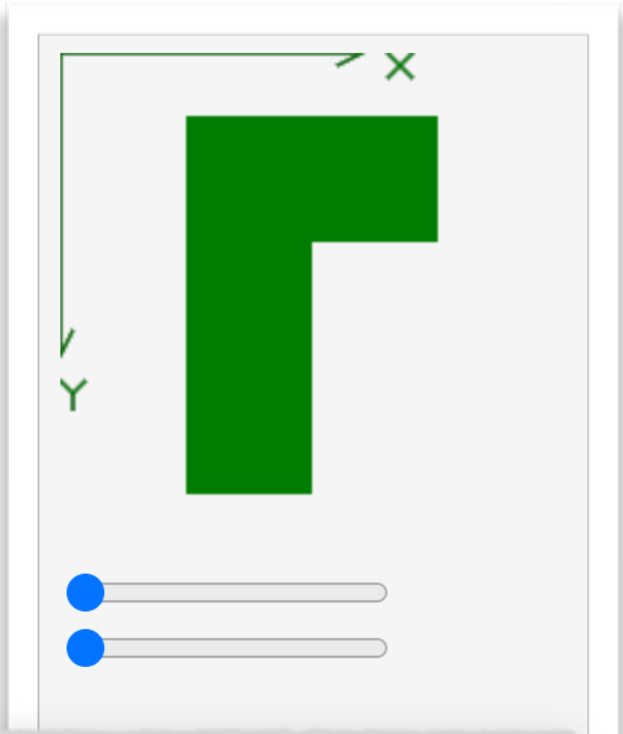
Additional features (JavaScrip

jsbin.com/fesukexori

Week2/Demo2

demo.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple demonstration of slider interface</title>
  </head>
  <body>
    <canvas id="myCanvas"
        width="400" height="400">
    </canvas>
    <br/>
    <input id="slider1" type="range" min="0" max="100" />
    <br/>
    <input id="slider2" type="range" min="0" max="100" />
    <script src="demo.js" id="module"></script>
  </body>
</html>
```



Transforms (we'll get into this in much more detail next week …)

demo.js

```
function setup() {
  var canvas = document.getElementById('myCanvas');
  var slider1 = document.getElementById('slider1');
  slider1.value = 0;
  var slider2 = document.getElementById('slider2');
  slider2.value = 0;
  function draw() {
    var context = canvas.getContext('2d');
    canvas.width = canvas.width;
    // use the sliders to get various parameters
    var dx = slider1.value;
    var dy = slider2.value;

    function DrawLshape(color) {
      context.beginPath();
      context.fillStyle = color;
      context.moveTo(50,25);context.lineTo(150,25);context.lineTo(150,75);
      context.lineTo(100,75);context.lineTo(100,175);context.lineTo(50,175);
      context.closePath();
      context.fill();
    }

    function DrawAxes(color) {
      context.strokeStyle=color;
      context.beginPath();
      // Axes
      context.moveTo(120,0);context.lineTo(0,0);context.lineTo(0,120);
      // Arrowheads
      context.moveTo(110,5);context.lineTo(120,0);context.lineTo(110,-5);
      context.moveTo(5,110);context.lineTo(0,120);context.lineTo(-5,110);
      // X-label
      context.moveTo(130,0);context.lineTo(140,10);
      context.moveTo(130,10);context.lineTo(140,0);
      // Y-label
      context.moveTo(0,130);context.lineTo(5,135);context.lineTo(10,130);
      context.moveTo(5,135);context.lineTo(5,142);

      context.stroke();
     }

    // make sure you understand these

    DrawAxes("black");
    context.save();
    context.translate(dx,dy);
    DrawAxes("green");
    DrawLshape("green");
    context.restore();
  }
  slider1.addEventListener("input",draw);
  slider2.addEventListener("input",draw);
  draw();
}
window.onload = setup;
```

Additional examples (flash preview)

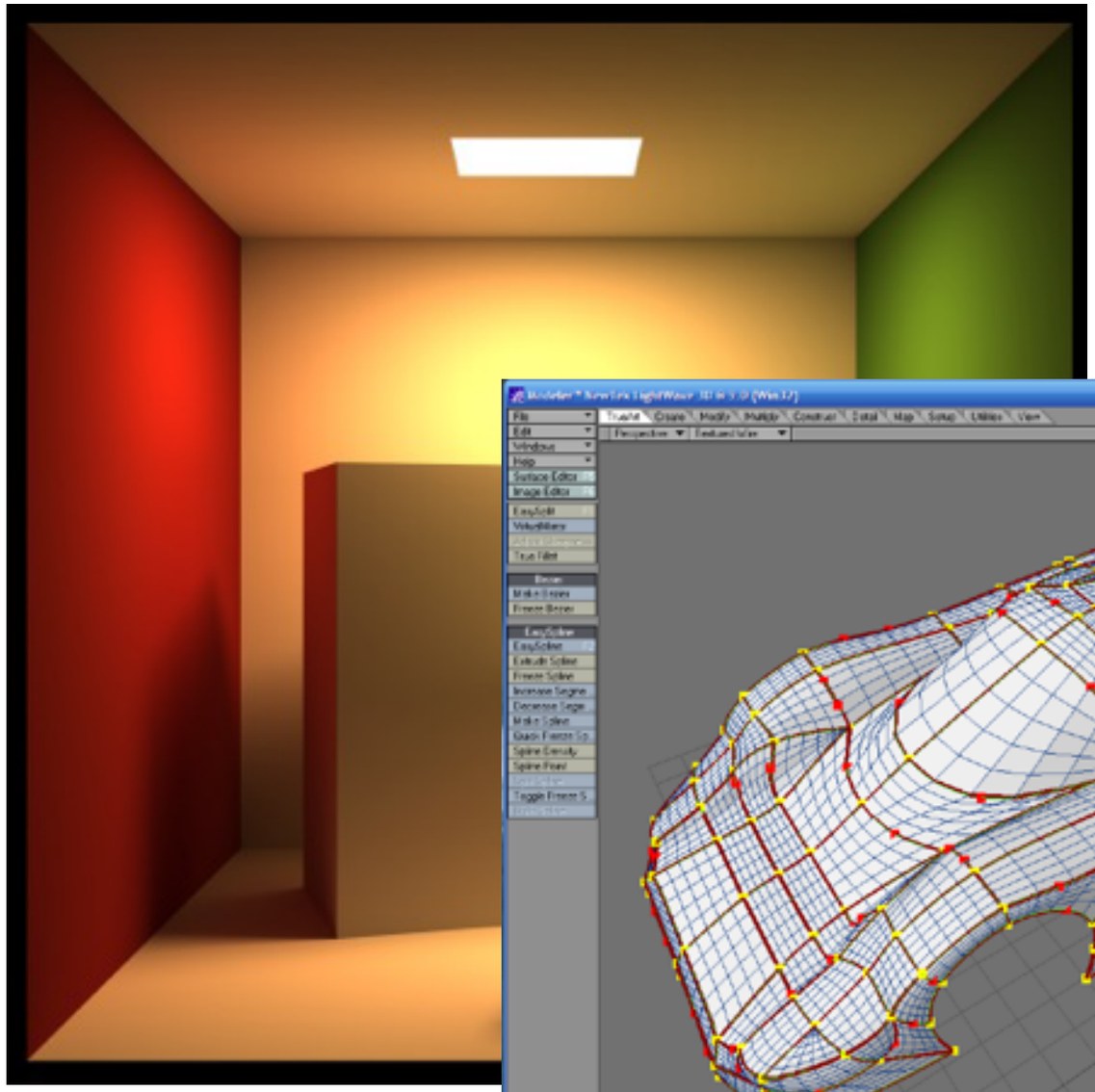
jsbin.com/zewesapifa

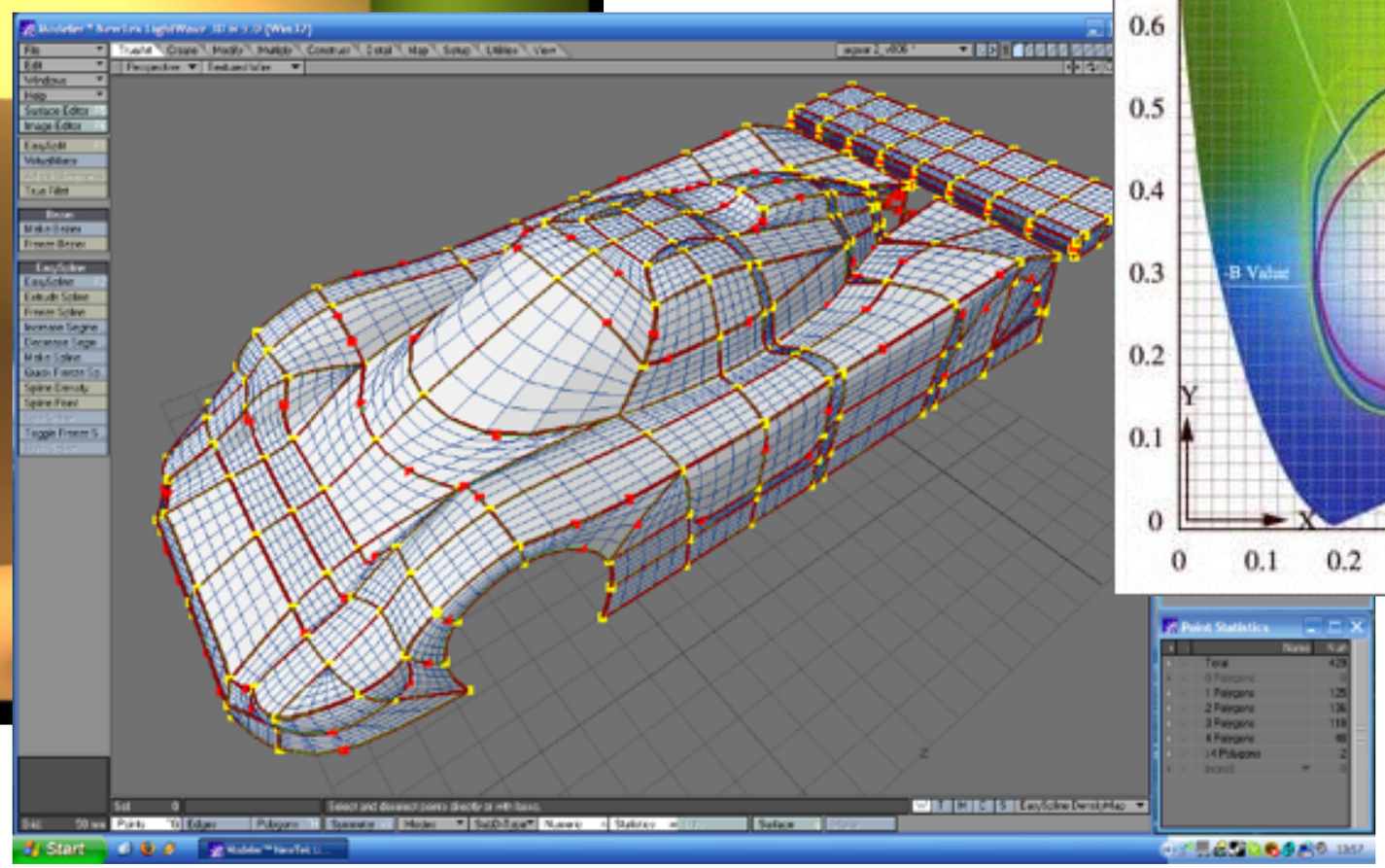
- Simple animation
  - Specify what action (drawing) is to be taken when the browser is ready to "refresh" its visual contents
  - Creates a recurrent action trigger, that we can use to create an animation

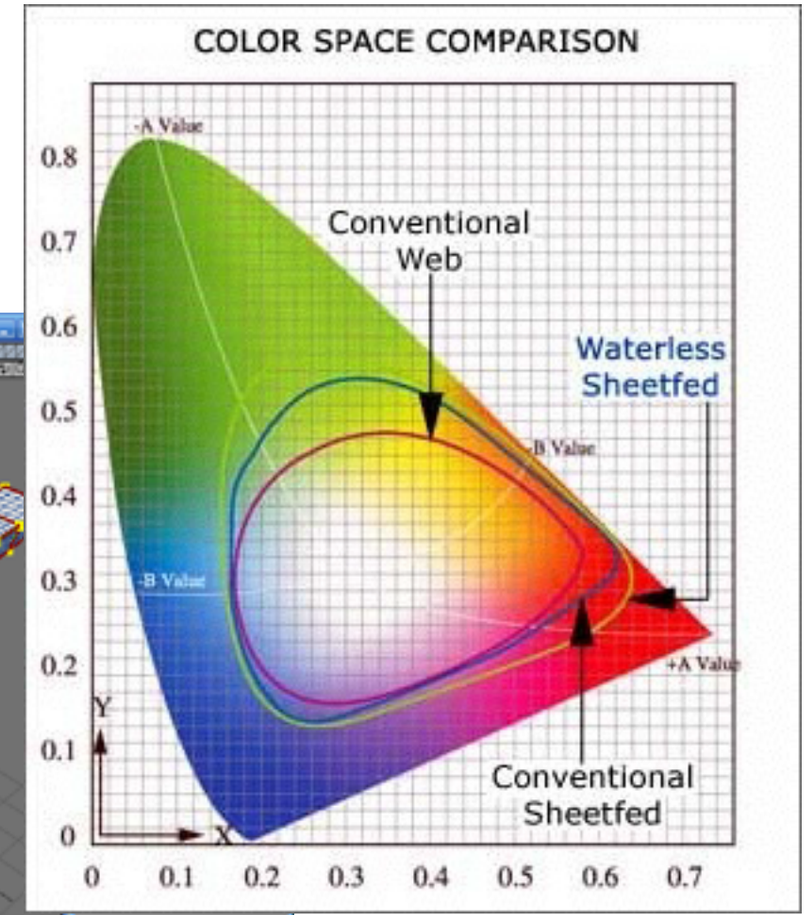
Additional examples (flash preview)

jsbin.com/wovupusife

- Hierarchical modeling
  - Shapes subordinate to the placement of other shapes
  - Uses several coordinate frames for drawing
  - Nested hierarchy of transforms
  - Illustration of the Canvas transform stack
  - (all of these to be discussed next week)







Lecture 3 : More features in Canvas 2D drawing (input elements, events/triggers, intro to transforms)

Thursday September 16th 2020