Understanding and Optimizing Web Graphics

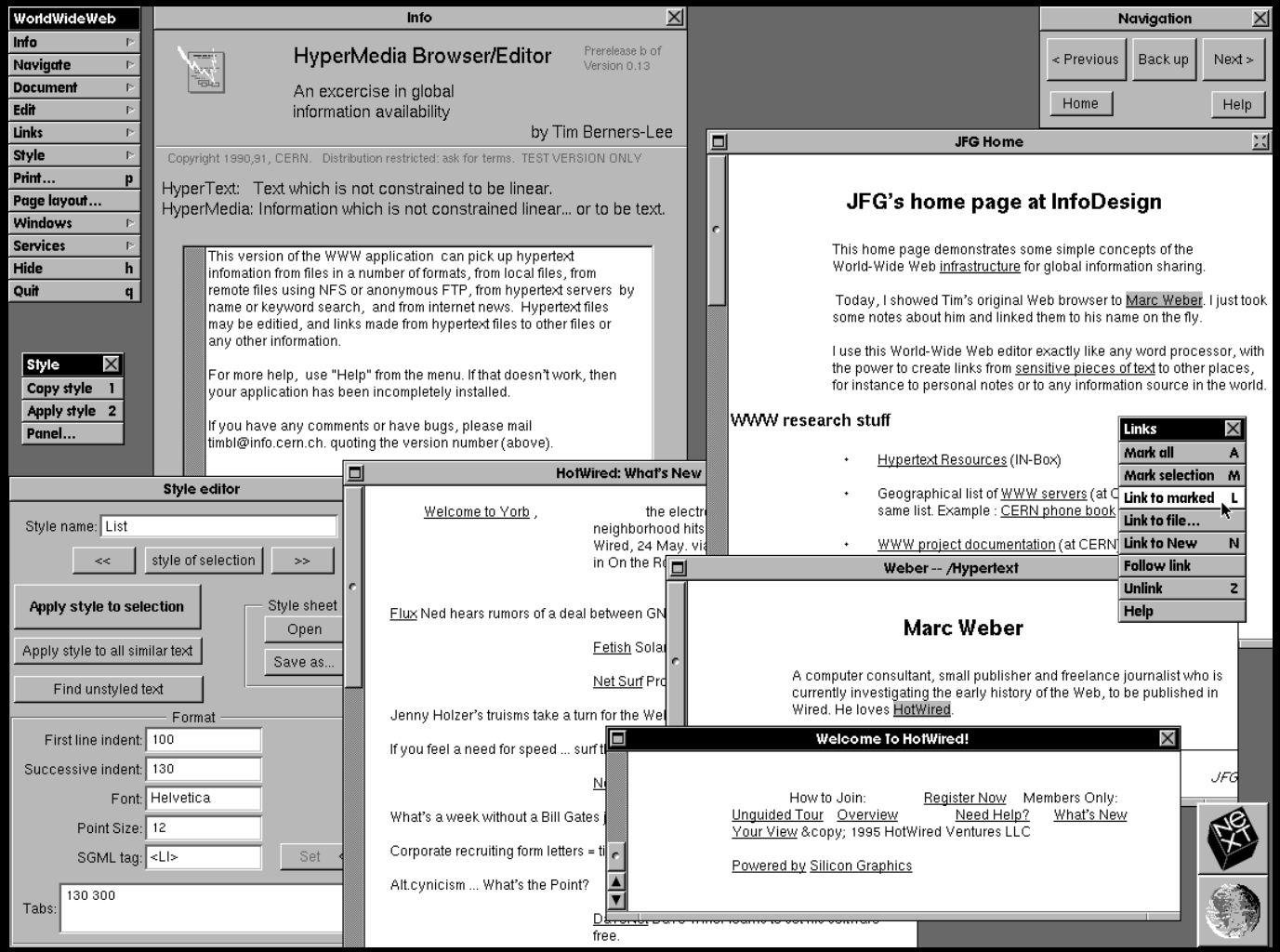
Session 508

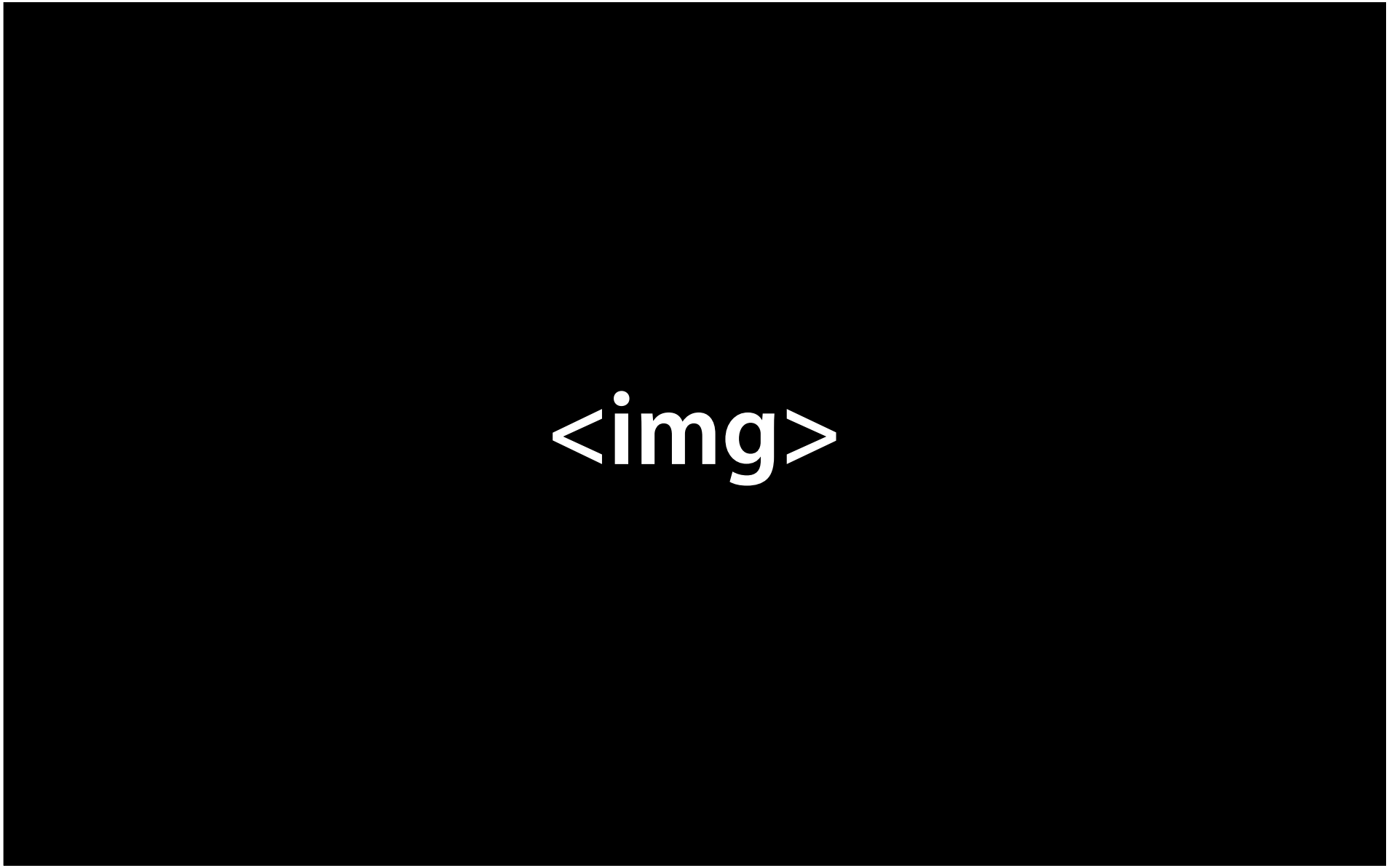
Dean Jackson
Apple

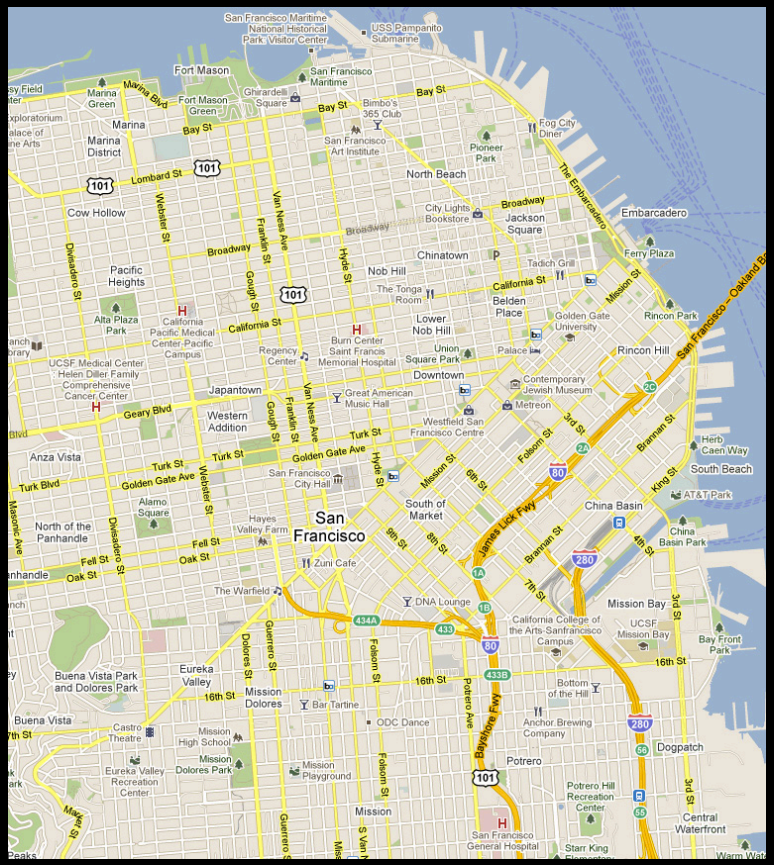

Pretty pictures
With Dean

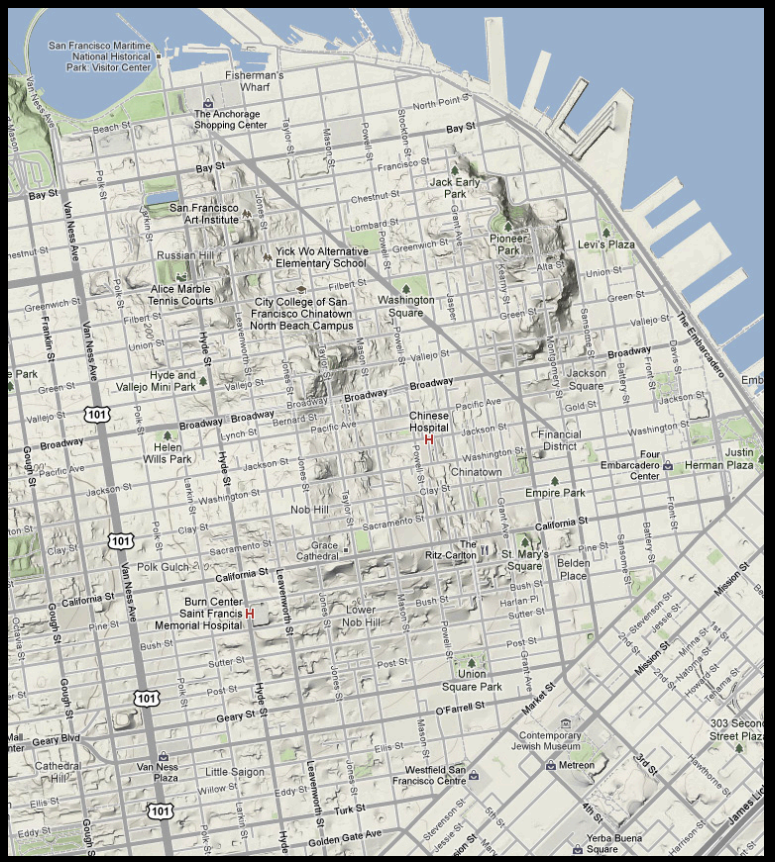

Apple – Mac
http://www.apple.com/mac/
Google
Store
Mac
iPod
iPhone
iPad
iTunes
Support
MacBook
MacBook Pro
MacBook Air
iMac
Mac mini
Mac Pro
Mac OS X
Mac
Applications
Accessories
Server
The ultimate all-in-one goes all out.
The new iMac now features a quad-core processor in every model, up to 3x faster graphics, Thunderbolt, and a FaceTime HD camera.
Starting at $1199.
The new MacBook Pro.
State-of-the-art processors. All-new graphics. Breakthrough high-speed I/O.
The new MacBook Air
The next generation of MacBooks.
The Mac App Store. Now open.
More than a thousand apps for your Mac — from your Mac.

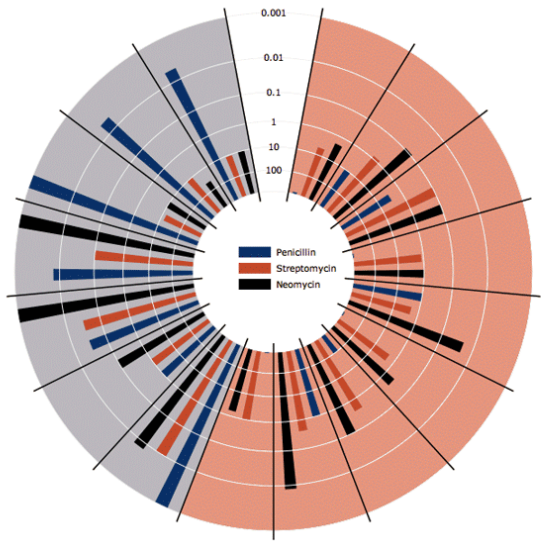

Apple – iCloud – The new way to store and access your content.
http://www.apple.com/icloud/
Google
Store
Mac
iPod
iPhone
iPad
iTunes
Support
iCloud
Overview
What is iCloud?
Features
Notify Me
Coming this fall
iCloud
This is the cloud the way it should be: automatic and effortless. iCloud is seamlessly integrated into your apps, so you can access your content on all your devices. And it's free with iOS 5.
Learn more
Your documents on all your devices.
What is iCloud?
Built right into your apps.
Watch the keynote video.

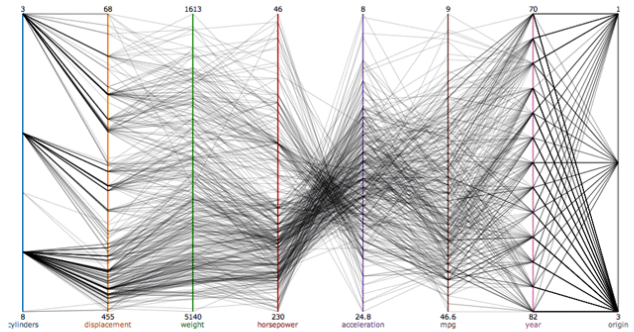

WorldWideWeb
Info
Navigate
Document
Edit
Links
Style
Print... p
Page layout...
Windows
Services
Hide h
Quit q
Style
Copy style 1
Apply style 2
Panel...
Info
HyperMedia Browser/Editor
Prerelease b of Version 0.13
An excercise in global information availability
by Tim Berners-Lee
Copyright 1990,91, CERN. Distribution restricted: ask for terms. TEST VERSION ONLY
HyperText: Text which is not constrained to be linear.
HyperMedia: Information which is not constrained linear... or to be text.
This version of the WWW application can pick up hypertext infomation from files in a number of formats, from local files, from remote files using NFS or anonymous FTP, from hypertext servers by name or keyword search, and from internet news. Hypertext files may be editied, and links made from hypertext files to other files or any other information.
For more help, use "Help" from the menu. If that doesn't work, then your application has been incompletely installed.
If you have any comments or have bugs, please mail timbl@info.cern.ch. quoting the version number (above).
Navigation
< Previous
Back up
Next >
Home
Help
JFG Home
JFG's home page at InfoDesign
This home page demonstrates some simple concepts of the World-Wide Web infrastructure for global information sharing.
Today, I showed Tim's original Web browser to Marc Weber. I just took some notes about him and linked them to his name on the fly.
I use this World-Wide Web editor exactly like any word processor, with the power to create links from sensitive pieces of text to other places, for instance to personal notes or to any information source in the world.
WWW research stuff
Hypertext Resources (IN-Box)
Geographical list of WWW servers (at C
same list. Example : CERN phone book
WWW project documentation (at CERN
Links
Mark all A
Mark selection M
Link to marked L
Link to file...
Link to New N
Follow link
Unlink Z
Help
Style editor
Style name: List
<<
style of selection
>>
Apply style to selection
Apply style to all similar text
Find unstyled text
Style sheet
Open
Save as...
Format
First line indent: 100
Successive indent: 130
Font: Helvetica
Point Size: 12
SGML tag: <LI>
Set
Tabs: 130 300
HotWired: What's New
Welcome to Yorb ,
the electr
neighborhood hits
Wired, 24 May. via
in On the R
Flux Ned hears rumors of a deal between GN
Fetish Sola
Net Surf Pr
Jenny Holzer's truisms take a turn for the We
If you feel a need for speed ... surf t
What's a week without a Bill Gates
Corporate recruiting form letters = t
Alt.cynicism ... What's the Point?
free.
Weber -- /Hypertext
Marc Weber
A computer consultant, small publisher and freelance journalist who is currently investigating the early history of the Web, to be published in Wired. He loves HotWired.
Welcome To HotWired!
How to Join:
Register Now
Members Only:
Unguided Tour
Overview
Need Help?
What's New
Your View
© 1995 HotWired Ventures LLC
Powered by Silicon Graphics

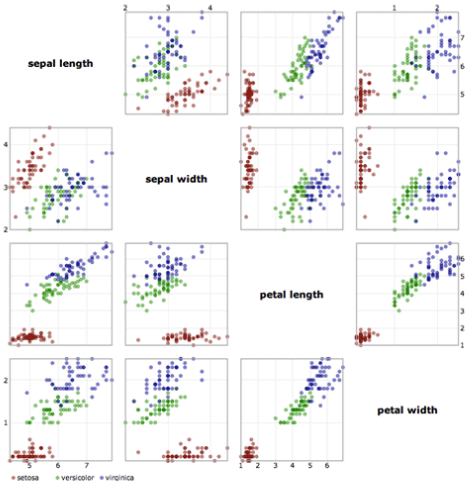

<img>

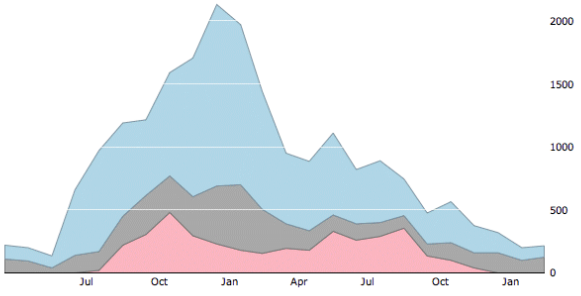

San Francisco Maritime National Historical Park Visitor Center
USS Pampanito Submarine
Fort Mason
San Francisco Maritime
Marina Green
Fort Mason Green
Ghirardelli Square
Bimbo's 365 Club
Marina
Marina District
San Francisco Art Institute
Pioneer Park
Fog City Diner
North Beach
Cow Hollow
City Lights Bookstore
Jackson Square
Embarcadero
Ferry Plaza
Pacific Heights
Chinatown
Nob Hill
Tadich Grill
The Tonga Room
Belden Place
Golden Gate University
Rincon Park
Alta Plaza Park
California Pacific Medical Center Pacific Campus
Lower Nob Hill
Union Square Park
Palace
Rincon Hill
UCSF Medical Center Helen Diller Family Comprehensive Cancer Center
Regency Center
Burn Center Saint Francis Memorial Hospital
Contemporary Jewish Museum
Japantown
Great American Music Hall
Metreon
Geary Blvd
Western Addition
Westfield San Francisco Centre
Herb Caen Way
Anza Vista
Turk Blvd
Golden Gate Ave
San Francisco City Hall
South Beach
AT&T Park
Alamo Square
South of Market
China Basin
North of the Panhandle
Hayes Valley Farm
San Francisco
China Basin Park
Fell St
Oak St
Zuni Cafe
The Warfield
DNA Lounge
Mission Bay
California College of the Arts San Francisco Campus
UCSF Mission Bay
Bay Front Park
Buena Vista Park and Dolores Park
Eureka Valley
16th St
Mission Dolores
Bar Tartine
Bottom of the Hill
Buena Vista
Castro Theatre
ODC Dance
Anchor Brewing Company
Mission High School
Dogpatch
Eureka Valley Recreation Center
Mission Dolores Park
Mission Playground
Potrero
Potrero Hill Recreation Center
Mission
San Francisco General Hospital
Central Waterfront
Starr King

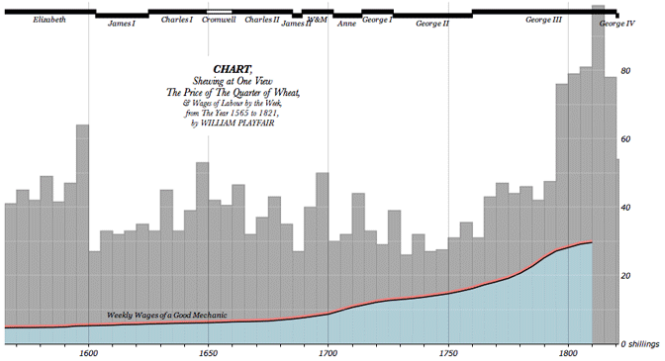

San Francisco Maritime National Historical Park Visitor Center
Fisherman's Wharf
The Anchorage Shopping Center
Bay St
Jack Early Park
San Francisco Art Institute
Pioneer Park
Levi's Plaza
Russian Hill
Yick Wo Alternative Elementary School
Alice Marble Tennis Courts
City College of San Francisco Chinatown North Beach Campus
Washington Square
Hyde and Vallejo Mini Park
Jackson Square
Chinese Hospital
Financial District
Helen Wills Park
Chinatown
Four Embarcadero Center
Herman Plaza
Empire Park
Nob Hill
Grace Cathedral
The Ritz-Carlton
St Mary's Square
Belden Place
Polk Gulch
Burn Center Saint Francis Memorial Hospital
Lower Nob Hill
Union Square Park
Cathedral Hill
Van Ness Plaza
Little Saigon
Westfield San Francisco Centre
Contemporary Jewish Museum
Metreon
Yerba Buena Square

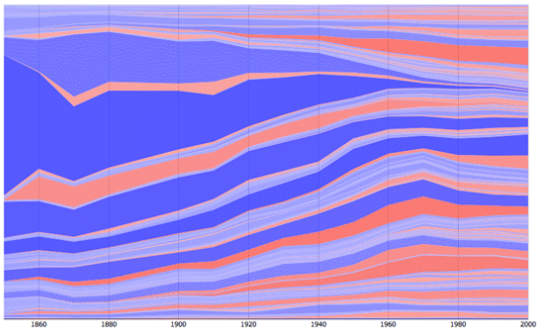

0.001
0.01
0.1
1
10
100
Penicillin
Streptomycin
Neomycin

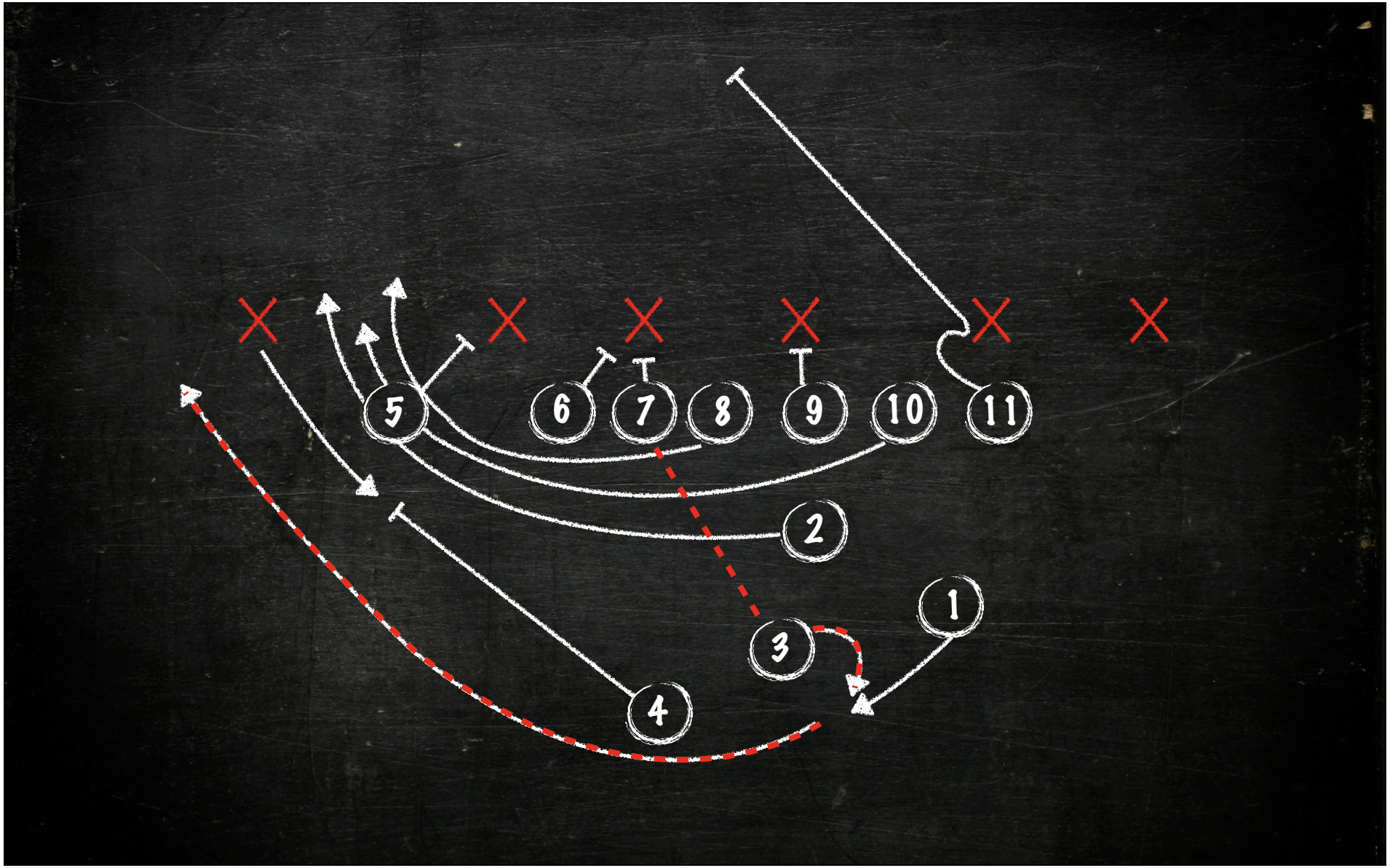

3
68
1613
46
8
9
70
1
8
cylinders
455
displacement
5140
weight
230
horsepower
24.8
acceleration
46.6
mpg
82
year
3
origin


sepal length
sepal width
petal length
petal width
setosa
versicolor
virginica


2000
1500
1000
500
0
Jul
Oct
Jan
Apr
Jul
Oct
Jan

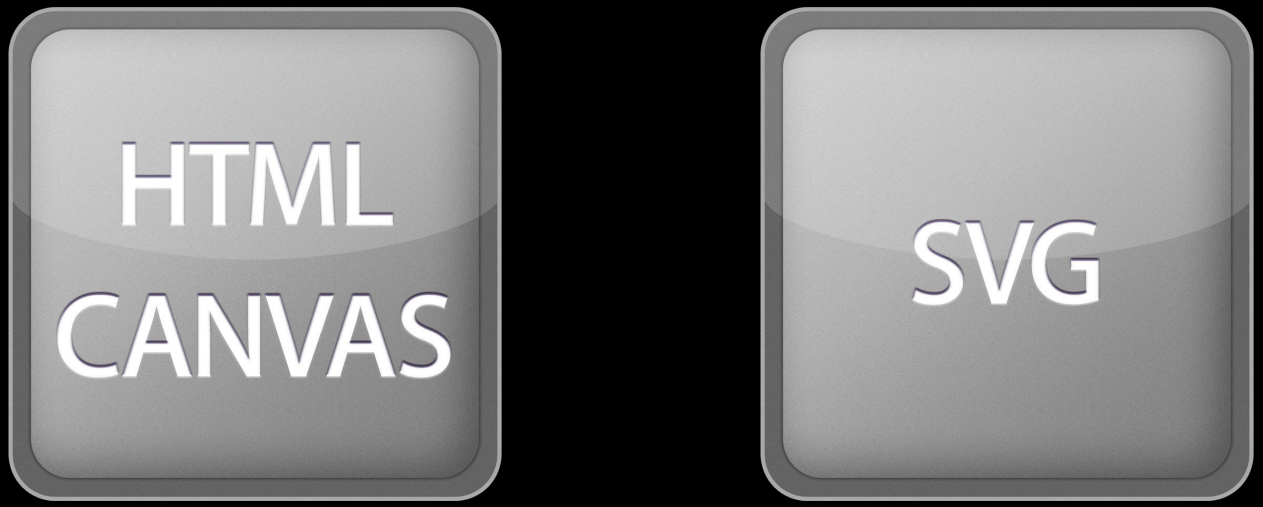

Elizabeth
James I
Charles I
Cromwell
Charles II
James II
W&M
Anne
George I
George II
George III
George IV
CHART,
Shewing at One View
The Price of The Quarter of Wheat,
& Wages of Labour by the Week,
from The Year 1565 to 1821,
by WILLIAM PLAYFAIR
Weekly Wages of a Good Mechanic
80
60
40
20
0 shillings
1600
1650
1700
1750
1800


1860
1880
1900
1920
1940
1960
1980
2000

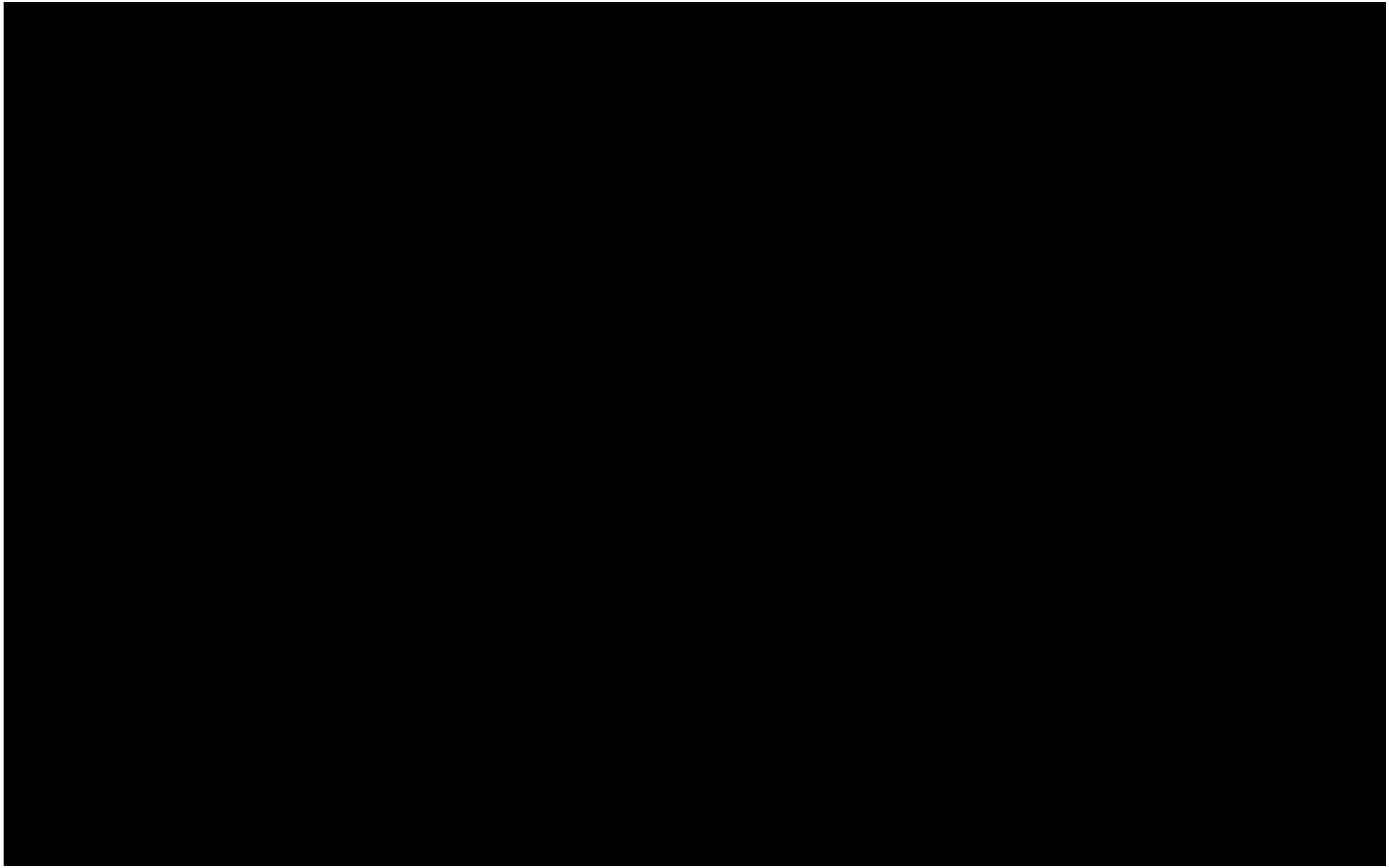

5
6
7
8
9
10
11
2
1
3
4


SCORE: 00485
LEVEL: 3
ALIENS LEFT: 07

What Technologies Are We Talking About?

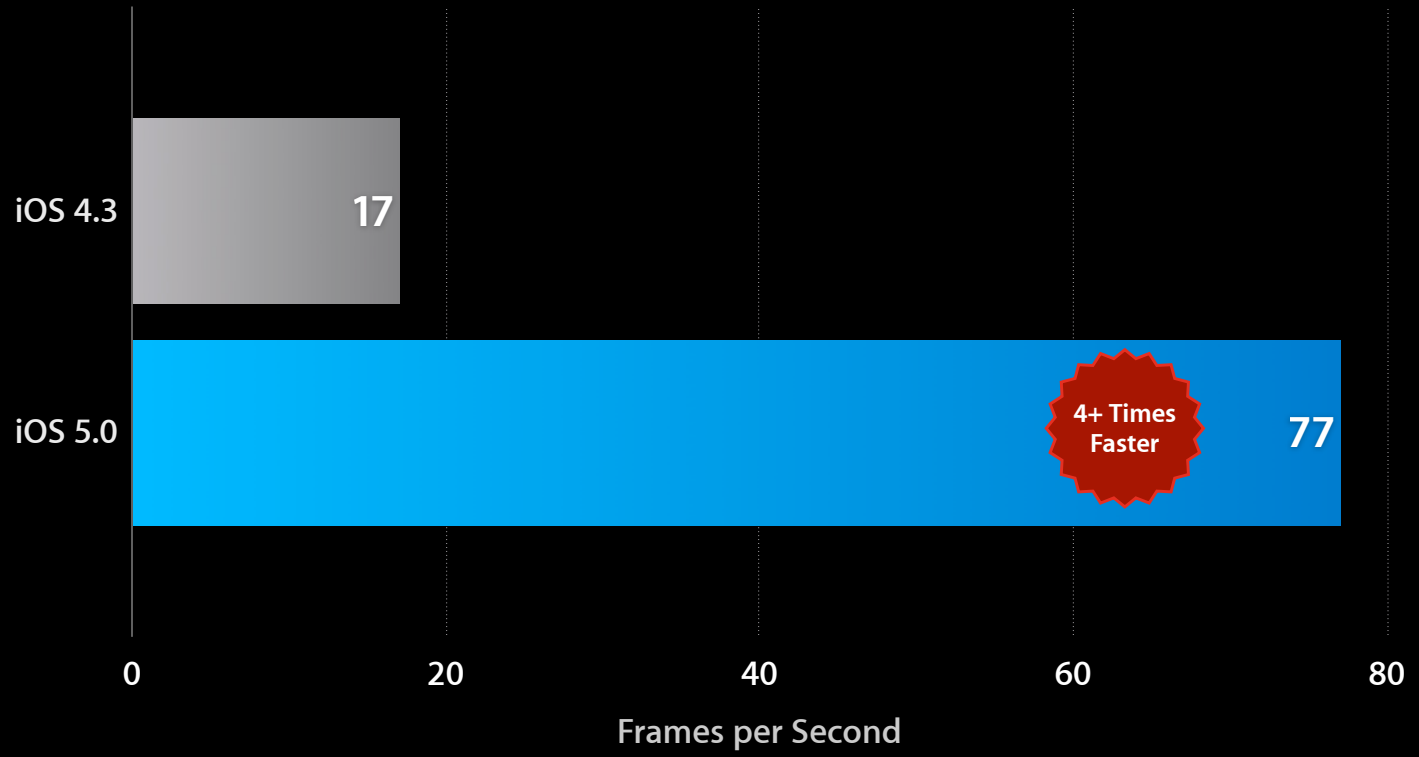


Why Talk About Graphics Now?

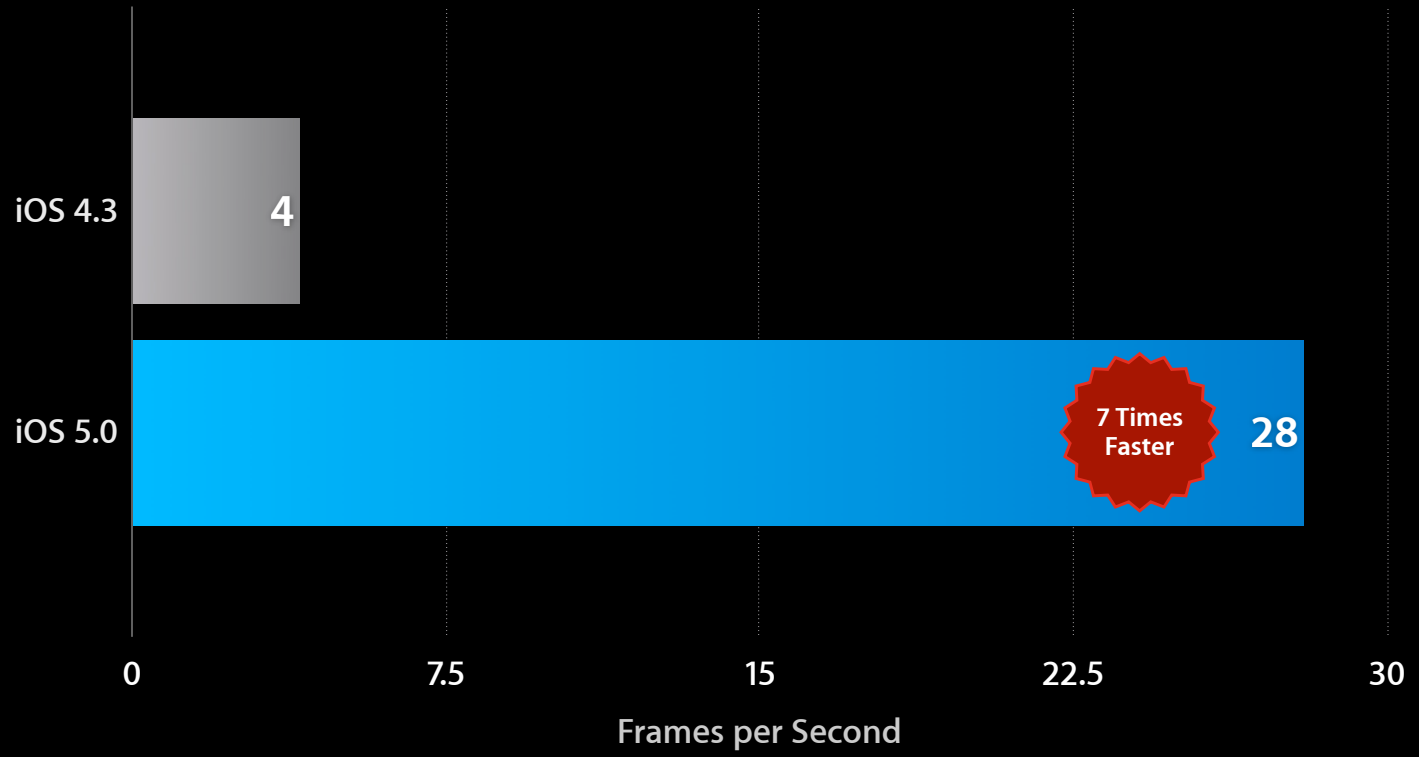


Demo

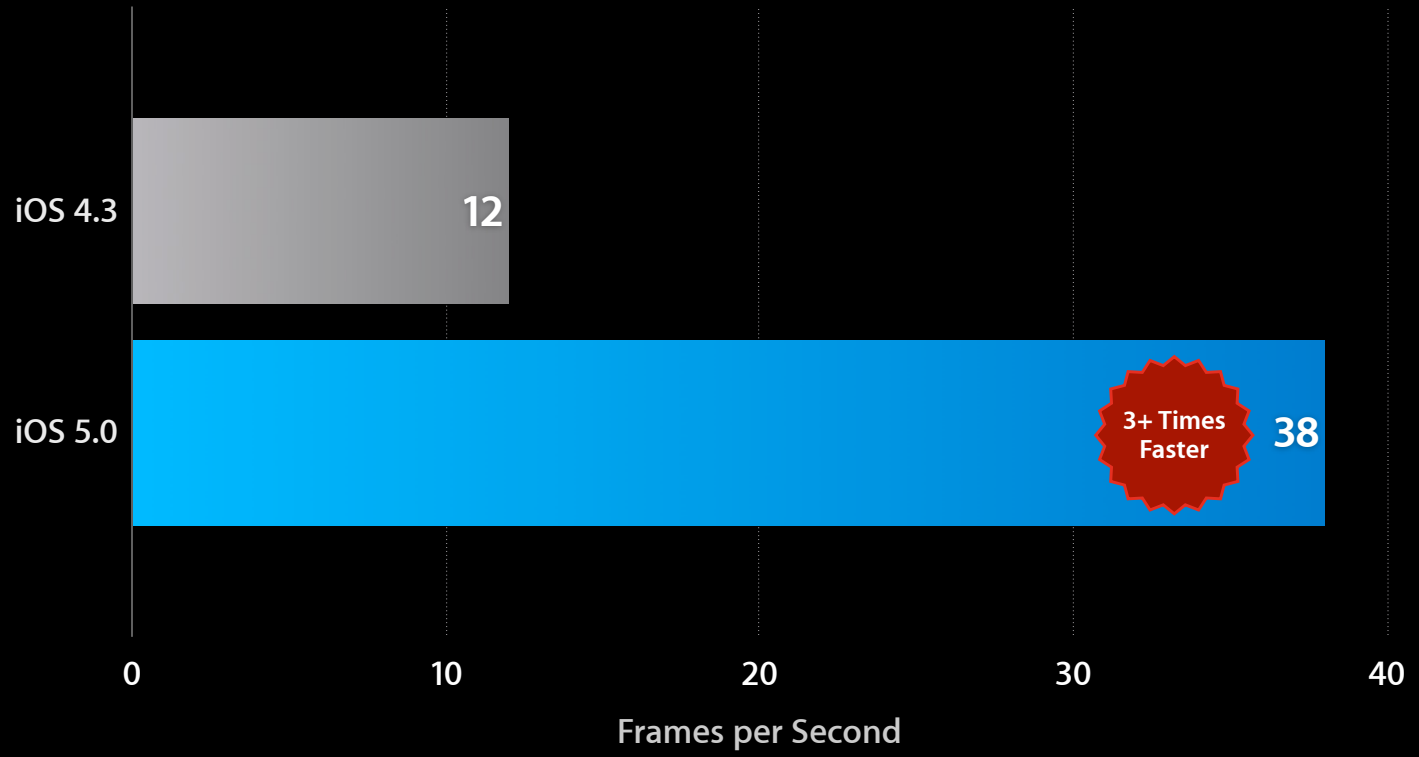
WWDC2011 Benchmark



Internet Explorer 9 Fish Tank Benchmark



“The Man In Blue” Benchmark



What Do You Need to Do?

Nothing!

Why Talk About Graphics Now?

What You'll Learn

1. Basic Concepts with Canvas and SVG
2. Animation and Interactivity
3. Tips and Special Effects

Basic Concepts

Canvas and SVG

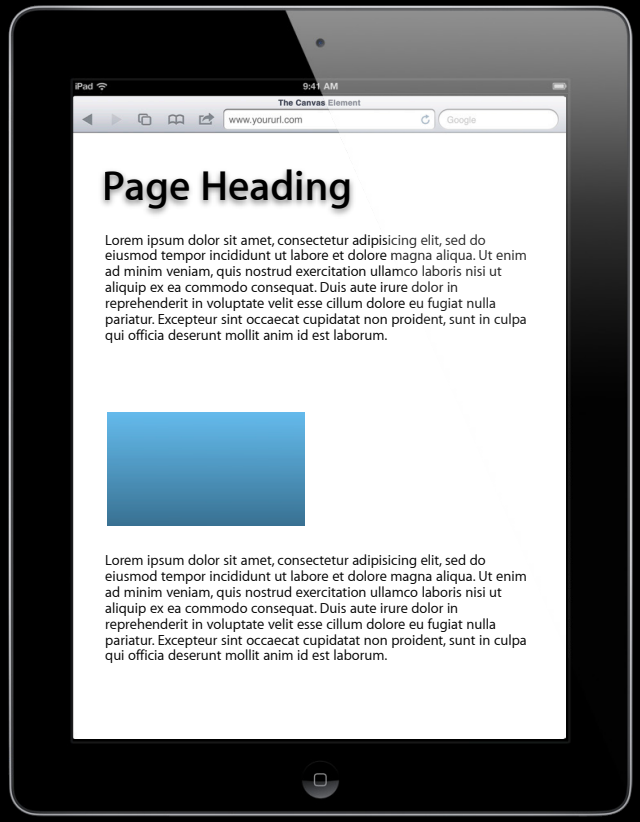
The HTML5 Canvas Element

<canvas>

The Canvas Element

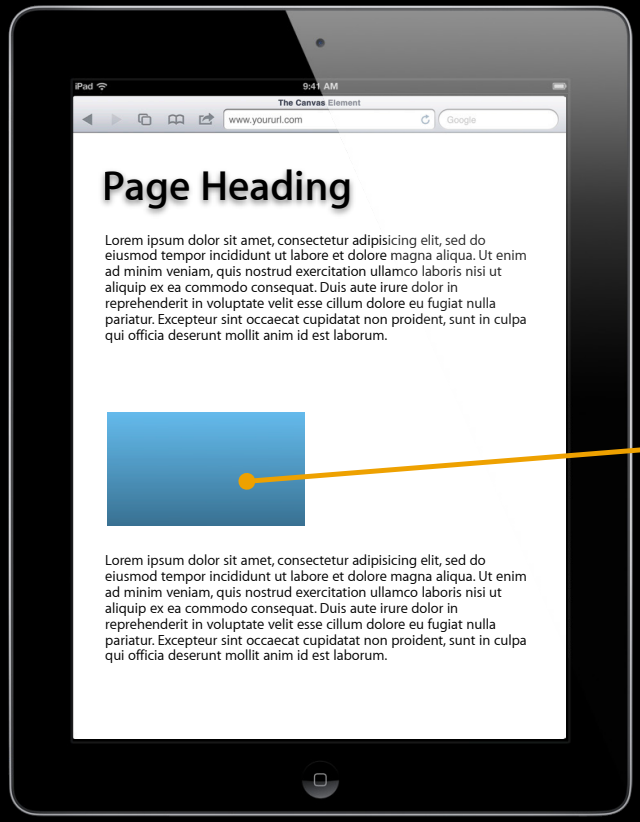
- Canvas is like a blank `<img>` with JavaScript API
- Instead of pixels coming from a file, you decide what is drawn in the image using JavaScript
- Each command is immediately rendered into the canvas

Using Canvas



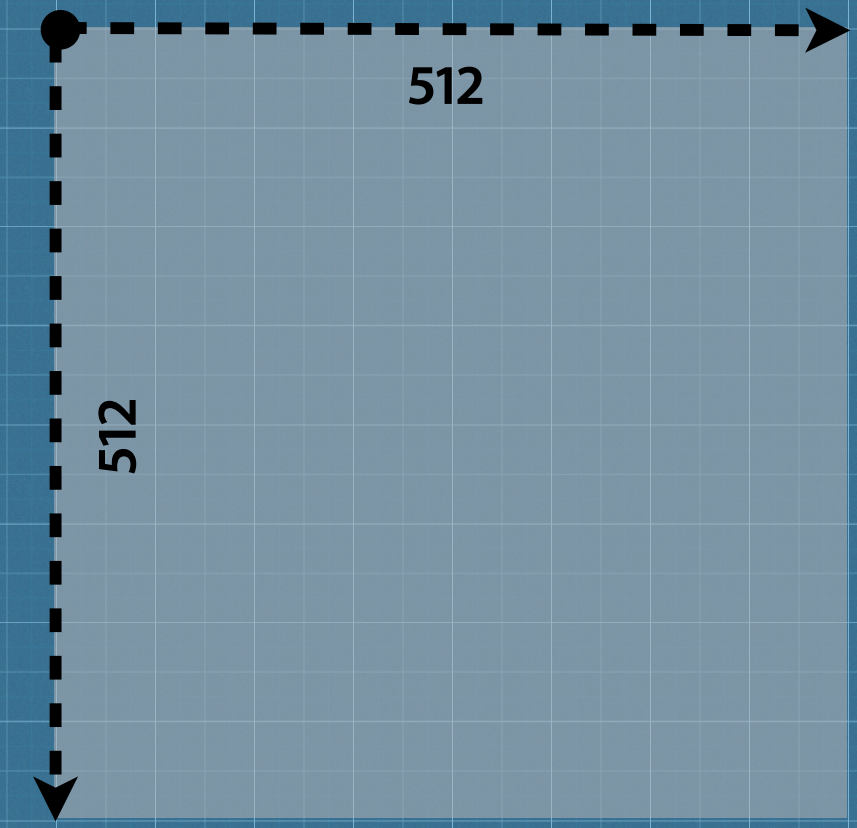

```
<html>
  <head> ... </head>
  <body>
    <h1>Page Heading</h1>

    <p>Lorem ipsum ...</p>

    <canvas id="picture1"
            width="400"
            height="300"/>

    <p>Lorem ipsum ...</p>

  </body>
</html>
```

Using Canvas



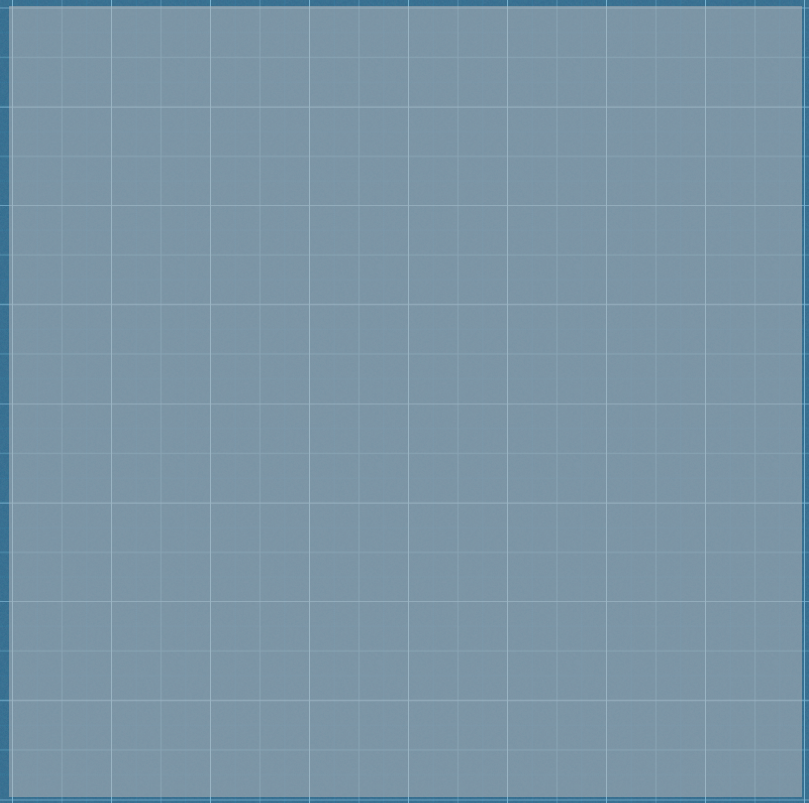

```
<html>
  <head> ... </head>
  <body>
    <h1>Page Heading</h1>

    <p>Lorem ipsum ...</p>

    <canvas id="picture1"
            width="400"
            height="300"/>

    <p>Lorem ipsum ...</p>

  </body>
</html>
```

Basic Drawing



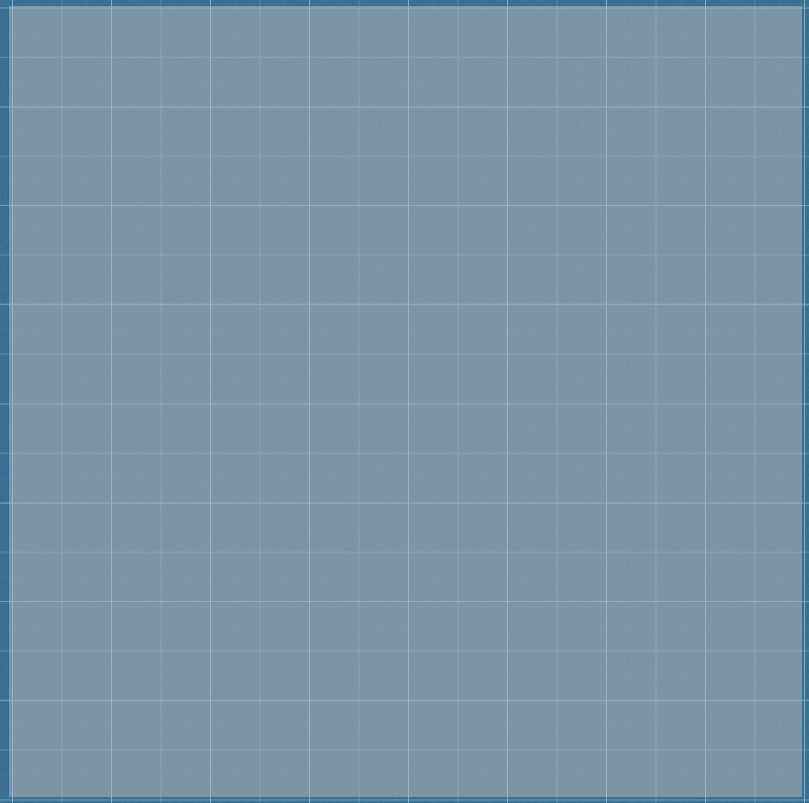
Basic Drawing

```
var ctx = canvas.getContext("2d");

ctx.clearRect(0, 0, 512, 512);

// draw shield background
ctx.beginPath();
ctx.moveTo(71, 460);
ctx.lineTo(30, 0);
ctx.lineTo(481, 0);
ctx.lineTo(440, 460);
ctx.lineTo(255, 512);
ctx.closePath();

ctx.fillStyle = "#E34F26";
ctx.fill();
```



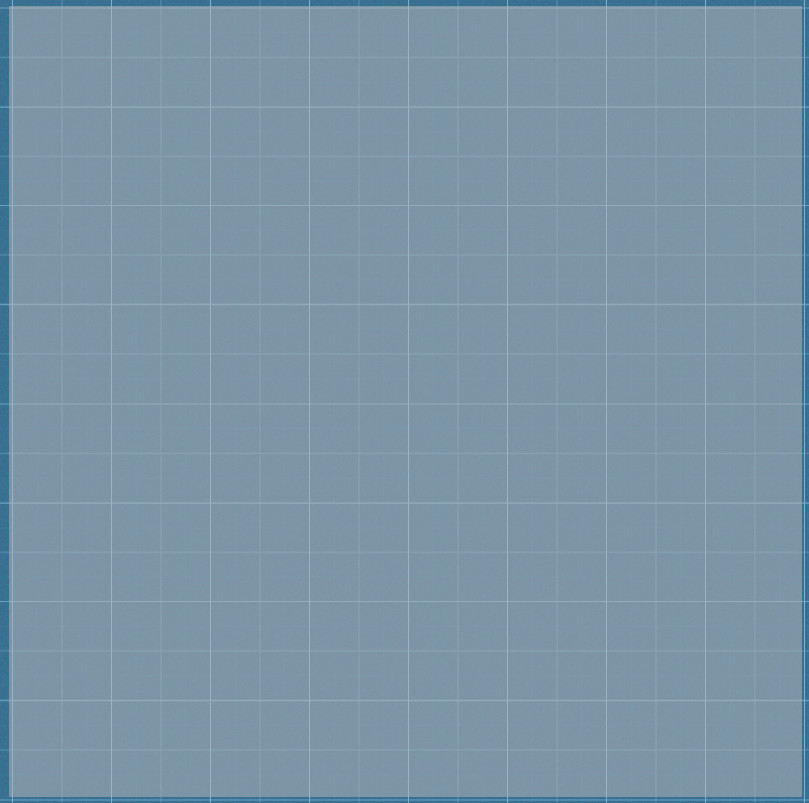
Basic Drawing

```
var ctx = canvas.getContext("2d");

ctx.clearRect(0, 0, 512, 512);

// draw shield background
ctx.beginPath();
ctx.moveTo(71, 460);
ctx.lineTo(30, 0);
ctx.lineTo(481, 0);
ctx.lineTo(440, 460);
ctx.lineTo(255, 512);
ctx.closePath();

ctx.fillStyle = "#E34F26";
ctx.fill();
```



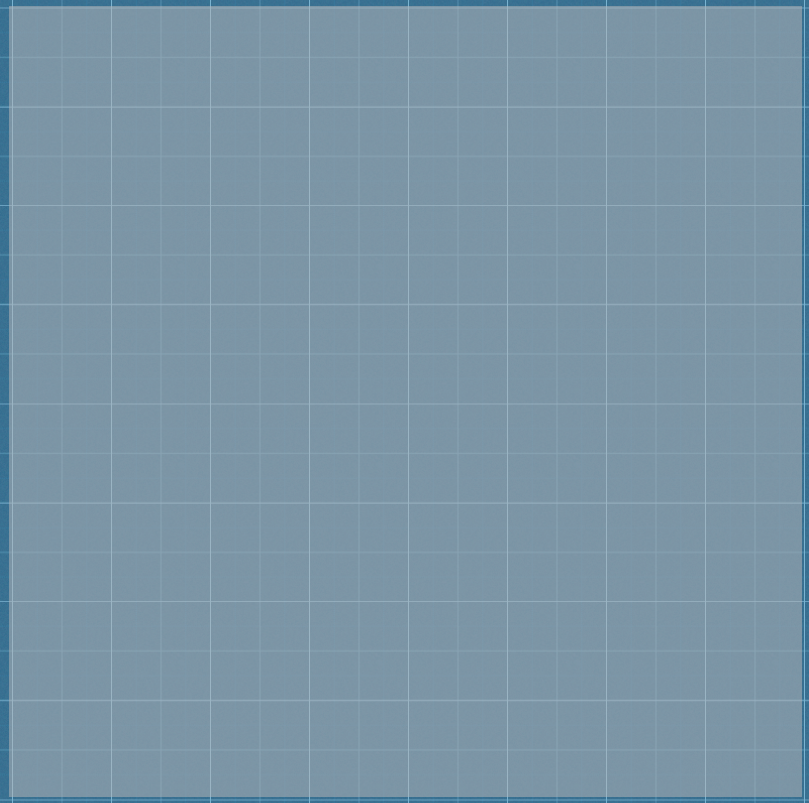
Basic Drawing

```
var ctx = canvas.getContext("2d");

ctx.clearRect(0, 0, 512, 512);

// draw shield background
ctx.beginPath();
ctx.moveTo(71, 460);
ctx.lineTo(30, 0);
ctx.lineTo(481, 0);
ctx.lineTo(440, 460);
ctx.lineTo(255, 512);
ctx.closePath();

ctx.fillStyle = "#E34F26";
ctx.fill();
```



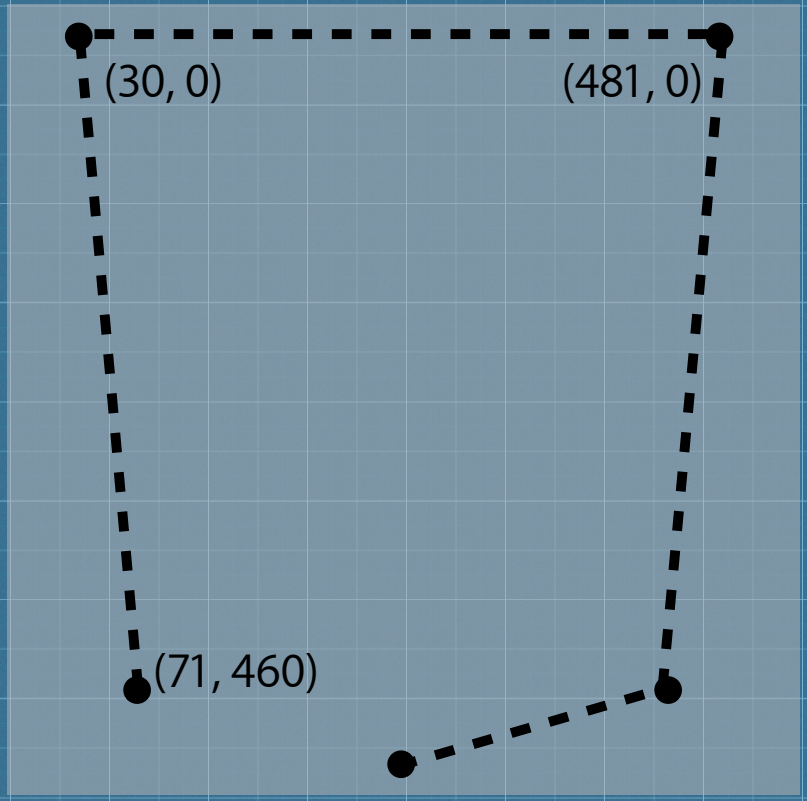
Basic Drawing

```
var ctx = canvas.getContext("2d");

ctx.clearRect(0, 0, 512, 512);

// draw shield background
ctx.beginPath();
ctx.moveTo(71, 460);
ctx.lineTo(30, 0);
ctx.lineTo(481, 0);
ctx.lineTo(440, 460);
ctx.lineTo(255, 512);
ctx.closePath();

ctx.fillStyle = "#E34F26";
ctx.fill();
```



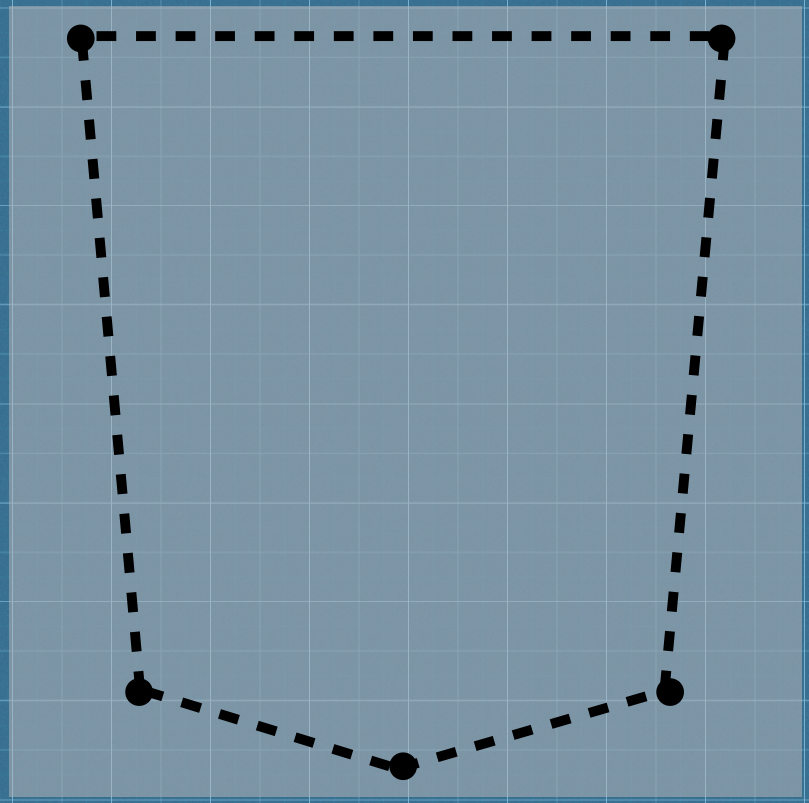
Basic Drawing

```
var ctx = canvas.getContext("2d");

ctx.clearRect(0, 0, 512, 512);

// draw shield background
ctx.beginPath();
ctx.moveTo(71, 460);
ctx.lineTo(30, 0);
ctx.lineTo(481, 0);
ctx.lineTo(440, 460);
ctx.lineTo(255, 512);
ctx.closePath();

ctx.fillStyle = "#E34F26";
ctx.fill();
```



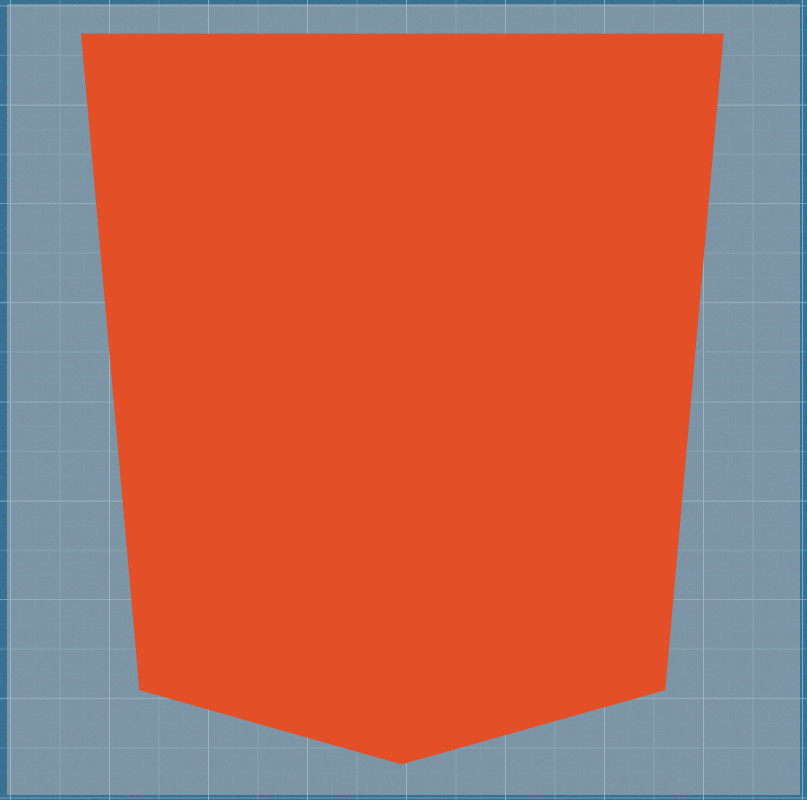
Basic Drawing

```
var ctx = canvas.getContext("2d");

ctx.clearRect(0, 0, 512, 512);

// draw shield background
ctx.beginPath();
ctx.moveTo(71, 460);
ctx.lineTo(30, 0);
ctx.lineTo(481, 0);
ctx.lineTo(440, 460);
ctx.lineTo(255, 512);
ctx.closePath();

ctx.fillStyle = "#E34F26";
ctx.fill();
```



Basic Drawing

```
var ctx = canvas.getContext("2d");

ctx.clearRect(0, 0, 512, 512);

// draw shield background
ctx.beginPath();
ctx.moveTo(71, 460);
ctx.lineTo(30, 0);
ctx.lineTo(481, 0);
ctx.lineTo(440, 460);
ctx.lineTo(255, 512);
ctx.closePath();

ctx.fillStyle = "#E34F26";
ctx.fill();
```



Basic Drawing

```
// draw number five
ctx.beginPath();
ctx.moveTo(181, 208);
ctx.lineTo(176, 150);
ctx.lineTo(392, 150);
ctx.lineTo(393, 138);
ctx.lineTo(396, 109);
...
ctx.lineTo(372, 372);
ctx.lineTo(385, 223);
ctx.lineTo(387, 208);
ctx.lineTo(371, 208);
ctx.closePath();

ctx.fillStyle = "#DBDBDB";
ctx.fill();
```

Basic Drawing

```
// draw shield highlight
ctx.beginPath();
ctx.moveTo(256, 472);
ctx.lineTo(405, 431);
ctx.lineTo(440, 37);
ctx.lineTo(256, 37);
ctx.closePath();

ctx.fillStyle =
    "rgba(255, 255, 255, 0.3)";
ctx.fill();
```

Basic Drawing

```
// draw shield highlight
ctx.beginPath();
ctx.moveTo(256, 472);
ctx.lineTo(405, 431);
ctx.lineTo(440, 37);
ctx.lineTo(256, 37);
ctx.closePath();

ctx.fillStyle =
    "rgba(255, 255, 255, 0.3)";
ctx.fill();
```

Using the Canvas

In HTML

```
<canvas id="html5logo"></canvas>
```

In CSS

```
h1 {
  background-image: -webkit-canvas("html5logo");
}

var ctx = document.getCSSCanvasContext("2d",
                              "html5logo", 500, 500);
```

Using the Canvas

In HTML

```
<canvas id="html5logo"></canvas>
```

In CSS

```
h1 {
  background-image: -webkit-canvas("html5logo");
}

var ctx = document.getCSSCanvasContext("2d",
                              "html5logo", 500, 500);
```

```
var canvas = document.querySelector("canvas");
var ctx = canvas.getContext("2d");

ctx.clearRect(0, 0, 512, 512);

// draw shield background
ctx.beginPath();
ctx.moveTo(71, 460);
ctx.lineTo(30, 0);
ctx.lineTo(481, 0);
ctx.lineTo(440, 460);
ctx.lineTo(255, 512);
ctx.closePath();

ctx.fillStyle = "#E34F26";
ctx.fill();

// draw number five
ctx.beginPath();
ctx.moveTo(181, 208);
ctx.lineTo(176, 150);
ctx.lineTo(392, 150);
ctx.lineTo(393, 138);
ctx.lineTo(396, 109);
ctx.lineTo(397, 94);
ctx.lineTo(114, 94);
ctx.lineTo(115, 109);
ctx.lineTo(129, 265);
ctx.lineTo(325, 265);
ctx.lineTo(318, 338);
ctx.lineTo(256, 355);
ctx.lineTo(192, 338);
ctx.lineTo(188, 293);
ctx.lineTo(132, 293);
ctx.lineTo(139, 382);
ctx.lineTo(256, 414);
ctx.lineTo(371, 382);
ctx.lineTo(372, 372);
ctx.lineTo(385, 223);
ctx.lineTo(387, 208);
ctx.lineTo(371, 208);
ctx.closePath();

ctx.fillStyle = "#DBDBDB";
ctx.fill();

// draw shield highlight
ctx.beginPath();
ctx.moveTo(256, 472);
ctx.lineTo(405, 431);
ctx.lineTo(440, 37);
ctx.lineTo(256, 37);
ctx.closePath();

ctx.fillStyle = "rgba(255, 255, 255, 0.3)";
ctx.fill();
```

SVG

Scalable Vector Graphics

Scalable Vector Graphics

- Markup format for graphics
- Vector-based, so perfect for resolution independence
- Lots of tool support

Creating the Graphic

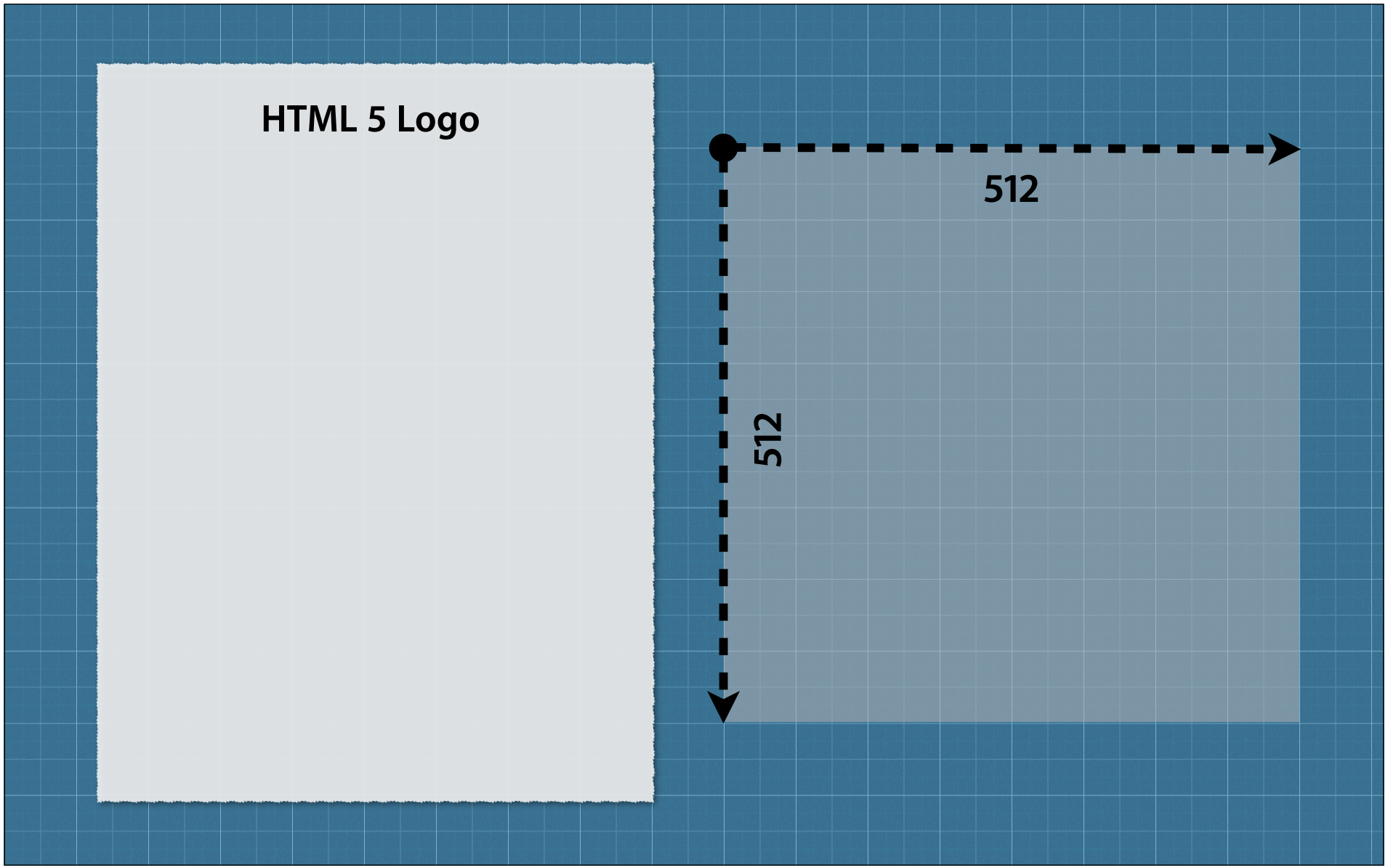
```
<svg>
  ... content goes here ...
</svg>
```

Creating the Graphic

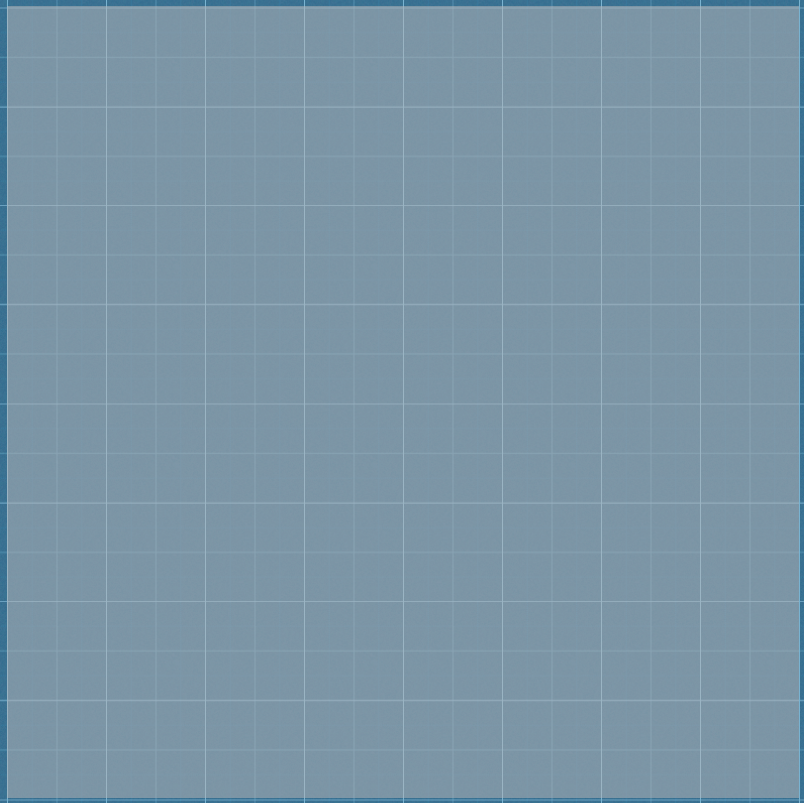
```
<svg viewBox="0 0 512 512">
  ... content goes here ...
</svg>
```

Creating the Graphic

```
<svg viewBox="0 0 512 512">
  ... content goes here ...
</svg>
```

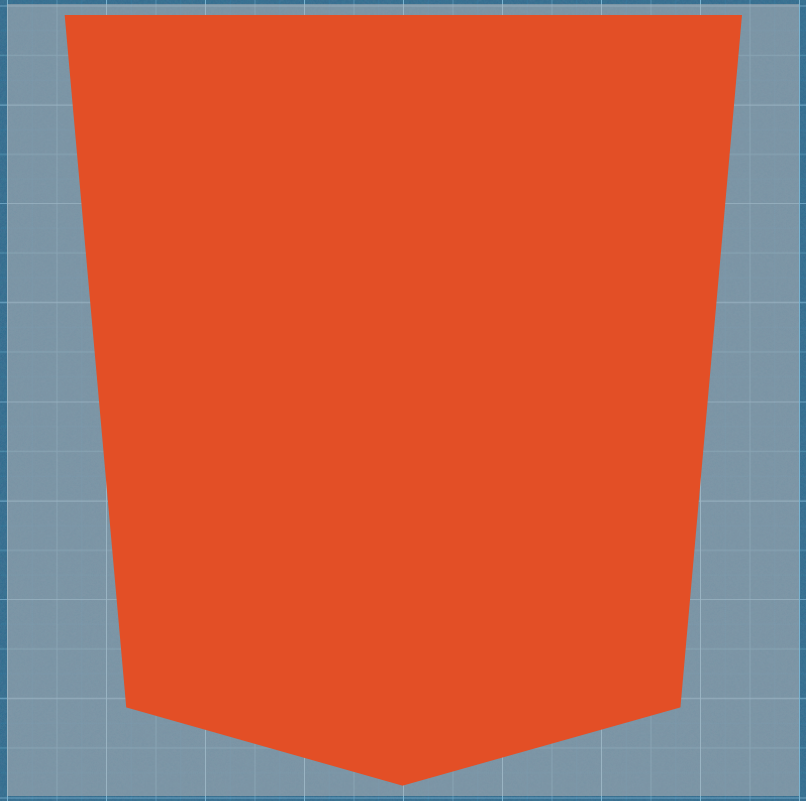

HTML 5 Logo
512
512

HTML 5 Logo



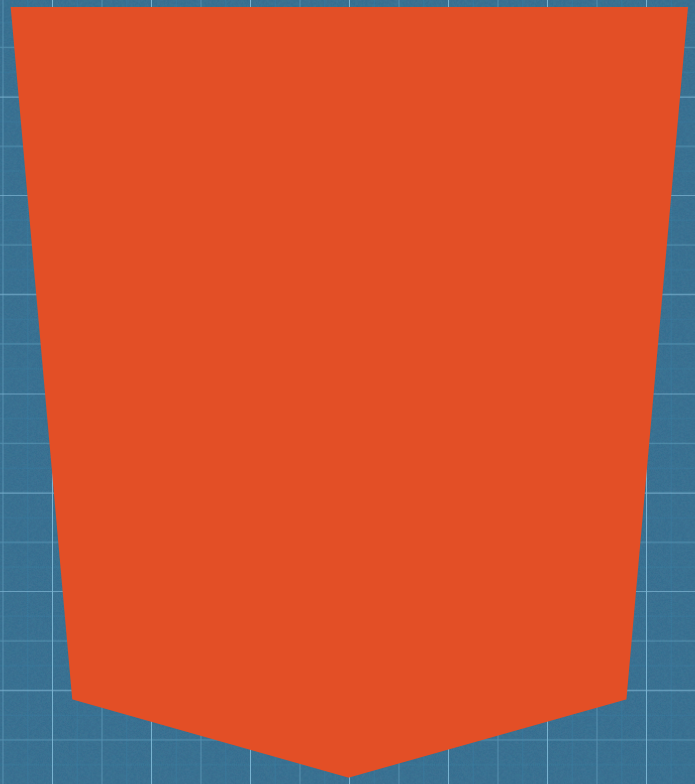
HTML 5 Logo

```
<polygon fill="#E34F26"
         points="71,460
                 30,0
                 481,0
                 440,460
                 255,512"/>
```



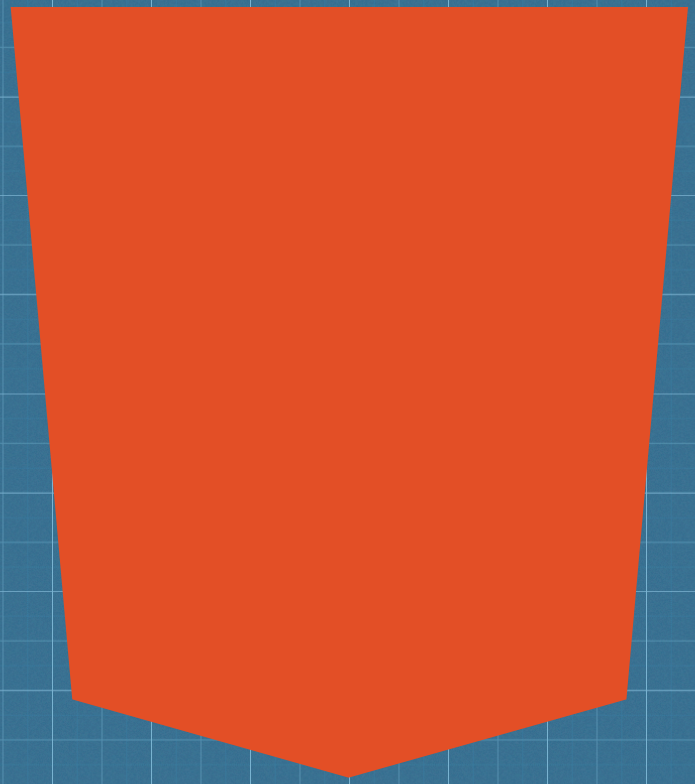
HTML 5 Logo

```
<polygon fill="#E34F26"
         points="71,460
                 30,0
                 481,0
                 440,460
                 255,512"/>
```



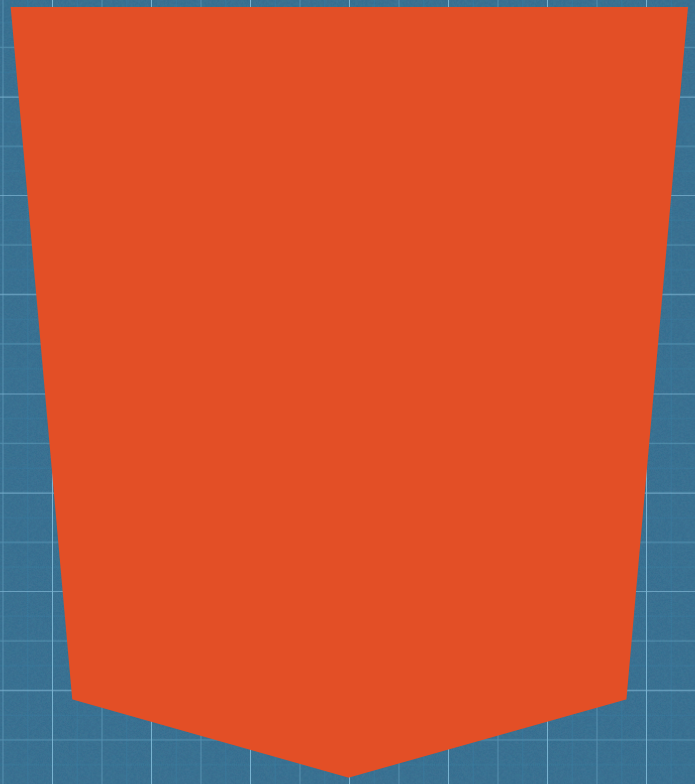
HTML 5 Logo

```
<polygon fill="#E34F26"
         points="71,460
                 30,0
                 481,0
                 440,460
                 255,512"/>
```



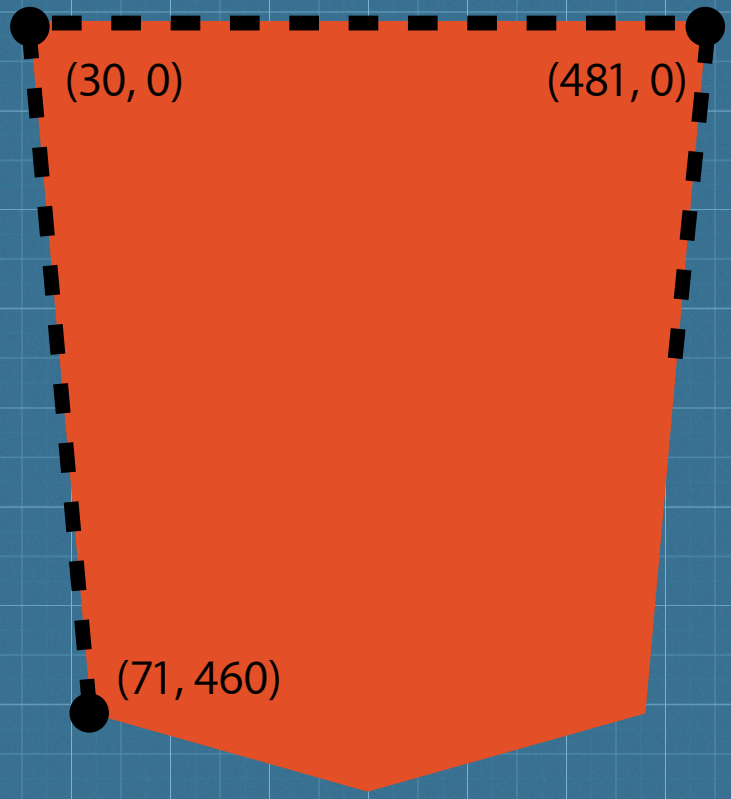
HTML 5 Logo

```
<polygon fill="#E34F26"
         points="71,460
                 30,0
                 481,0
                 440,460
                 255,512"/>
```



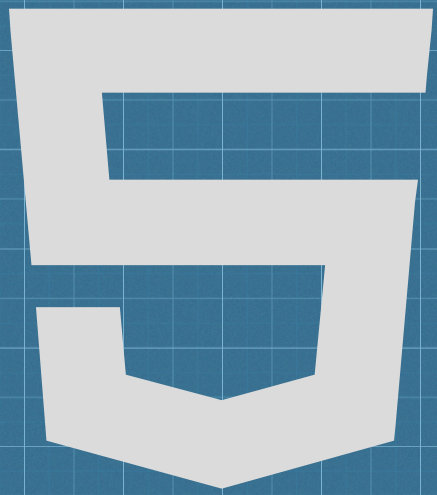
HTML 5 Logo

```
<polygon fill="#E34F26"
         points="71,460
                 30,0
                 481,0
                 440,460
                 255,512"/>
```



HTML 5 Logo

```
<polygon fill="#DBDBDB"
         points="181,208
                 176,150
                 392,150
                 393,138
                 396,109
                 397,94
                 114,94
                 115,109
                 129,265
                 325,265
                 318,338
                 256,355
                 192,338
                 188,293
                 132,293
                 139,382
                 256,414
                 371,382
                 372,372
                 385,223
                 387,208
                 371,208"/>
```



HTML 5 Logo

```
<polygon fill="#E34F26"
         points="71,460
                 30,0
                 481,0
                 440,460
                 255,512" />

<polygon fill="#DBDBDB"
         points="181,208
                 176,150
                 392,150
                 393,138
                 396,109
                 397,94
                 114,94 ...
```

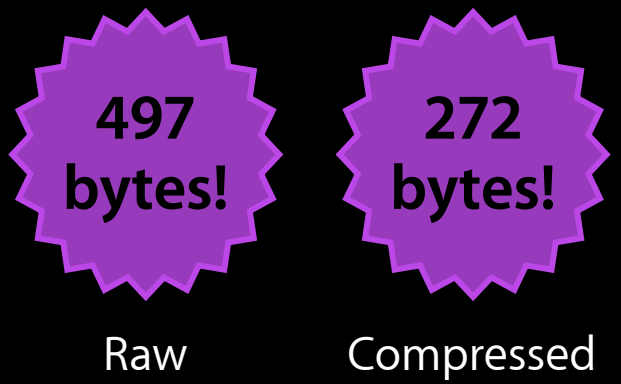
HTML 5 Logo

```
<polygon fill="white"
         opacity="0.3"
         points="256,472
                 405,431
                 440,37
                 256,37"/>
```

HTML 5 Logo

```
<polygon fill="white"
         opacity="0.3"
         points="256,472
                 405,431
                 440,37
                 256,37"/>
```

```
<svg viewBox="0 0 512 512">
  <polygon fill="#E34F26" points="71,460 30,0 481,0 440,460 255,512"/>
  <polygon fill="#DBDBDB" points="181,208 176,150 392,150 393,138 396,109 397,94
                                  114,94 115,109 129,265 325,265 318,338
                                  256,355 192,338 188,293 132,293 139,382
                                  256,414 371,382 372,372 385,223 387,208 371,208"/>
  <polygon fill="white" opacity="0.3" points="256,472 405,431 440,37 256,37"/>
</svg>
```



Using the Image

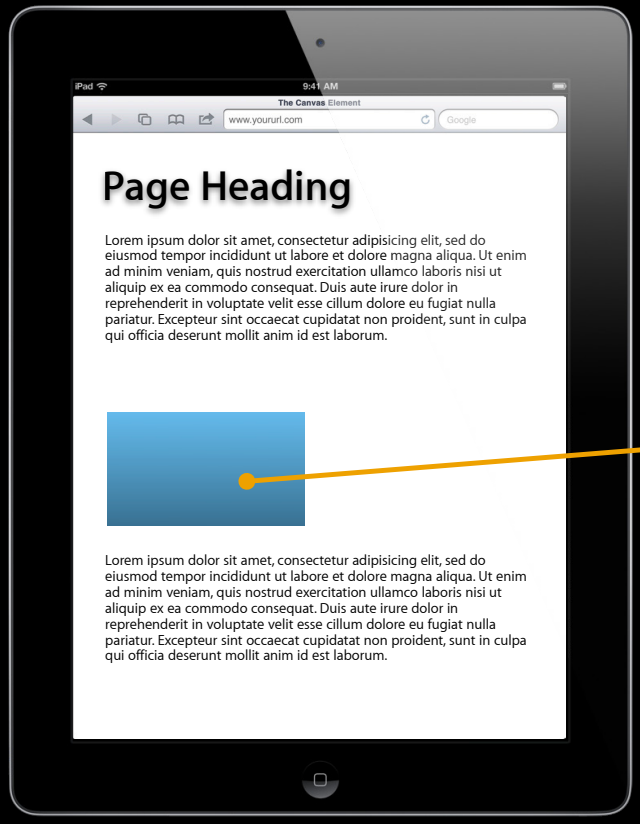
In HTML

```
<img src="html5logo.svg">
```

In CSS

```
h1 {
  background-image: url("html5logo.svg");
  background-size: 300px 300px;
}
```

Using SVG Inline




```
<html>
  <head> ... </head>
  <body>
    <h1>Page Heading</h1>

    <p>Lorem ipsum ...</p>

    <svg id="picture1">
      ...
    </svg>

    <p>Lorem ipsum ...</p>

  </body>
</html>
```

```svg
<svg viewBox="0 0 512 512">
  <polygon fill="#E34F26" points="71,460 30,0 481,0 440,460 255,512"/>
  <polygon fill="#DBDBDB" points="181,208 176,150 392,150 393,138 396,109 397,94
                                  114,94 115,109 129,265 325,265 318,338
                                  256,355 192,338 188,293 132,293 139,382
                                  256,414 371,382 372,372 385,223 387,208 371,208"/>
  <polygon fill="white" opacity="0.3" points="256,472 405,431 440,37 256,37"/>
</svg>
```

```svg
<svg viewBox="0 0 512 512">
  <polygon fill="#E34F26" points="71,460 30,0 481,0 440,460 255,512" />
  <polygon fill="#DBDBDB" points="181,208 176,150 392,150 393,138 396,109 397,94
                                  114,94 115,109 129,265 325,265 318,338
                                  256,355 192,338 188,293 132,293 139,382
                                  256,414 371,382 372,372 385,223 387,208 371,208" />
  <polygon fill="white" opacity="0.3" points="256,472 405,431 440,37 256,37" />
</svg>
```

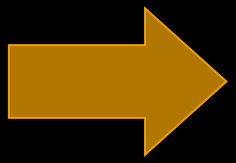
```svg
<svg viewBox="0 0 512 512">
  <style>
    .shield {
      fill: #E34F26;
    }

    .five {
      fill: #DBDBDB;
    }

    .highlight {
      fill: white;
      opacity: 0.3;
    }
  </style>
  <polygon class="shield" points="71,460 30,0 481,0 440,460 255,512" />
  <polygon class="five" points="181,208 176,150 392,150 393,138 396,109 397,94
                                114,94 115,109 129,265 325,265 318,338
                                256,355 192,338 188,293 132,293 139,382
                                256,414 371,382 372,372 385,223 387,208 371,208" />
  <polygon class="highlight" points="256,472 405,431 440,37 256,37" />
</svg>
```

```
<svg viewBox="0 0 512 512">
  <style>
    .shield {
      fill: #E34F26;
    }

    .five {
      fill: #DBDBDB;
    }

    .highlight {
      fill: white;
      opacity: 0.3;
    }
  </style>
  <polygon class="shield" points="71,460 30,0 481,0 440,460 255,512"/>
  <polygon class="five" points="181,208 176,150 392,150 393,138 396,109 397,94
                                114,94 115,109 129,265 325,265 318,338
                                256,355 192,338 188,293 132,293 139,382
                                256,414 371,382 372,372 385,223 387,208 371,208"/>
  <polygon class="highlight" points="256,472 405,431 440,37 256,37"/>
</svg>
```

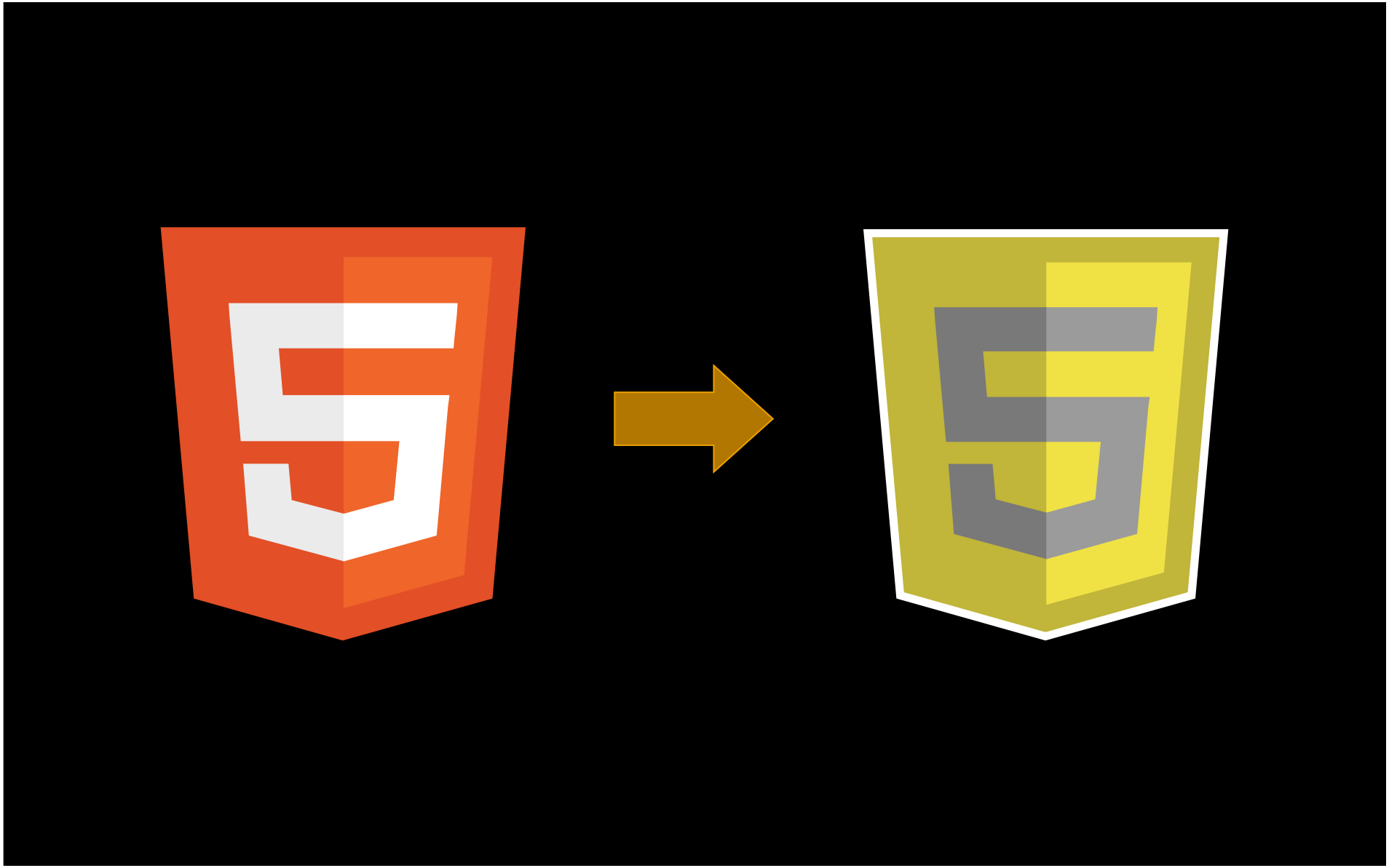
```svg
<svg viewBox="0 0 512 512">
  <style>
    .shield {
      fill: #E34F26;
    }

    .five {
      fill: #DBDBDB;
    }

    .highlight {
      fill: white;
      opacity: 0.3;
    }
  </style>
  <polygon class="shield" points="71,460 30,0 481,0 440,460 255,512"/>
  <polygon class="five" points="181,208 176,150 392,150 393,138 396,109 397,94
                                114,94 115,109 129,265 325,265 318,338
                                256,355 192,338 188,293 132,293 139,382
                                256,414 371,382 372,372 385,223 387,208 371,208"/>
  <polygon class="highlight" points="256,472 405,431 440,37 256,37"/>
</svg>
```

```svg
<svg viewBox="0 0 512 512">
  <style>
    .shield {
      fill: #E34F26;
    }

    .five {
      fill: #DBDBDB;
    }

    .highlight {
      fill: white;
      opacity: 0.3;
    }
  </style>
  <polygon class="shield" points="71,460 30,0 481,0 440,460 255,512"/>
  <polygon class="five" points="181,208 176,150 392,150 393,138 396,109 397,94
                                114,94 115,109 129,265 325,265 318,338
                                256,355 192,338 188,293 132,293 139,382
                                256,414 371,382 372,372 385,223 387,208 371,208"/>
  <polygon class="highlight" points="256,472 405,431 440,37 256,37"/>
</svg>
```

```css
.shield {
  fill: #E34F26;
}

.five {
  fill: #DBDBDB;
}

.highlight {
  fill: white;
  opacity: 0.3;
}
```



```css
.shield {
  fill: #C1B53A;
  stroke: white;
  stroke-width: 10px;
}

.five {
  fill: #9B9B9B;
}

.highlight {
  fill: #F0E145;
  opacity: 0.3;
}
```



	Canvas	SVG
Mode	Immediate	Retained
Basic Graphics	✓	✓
Background Image	✓	✓
Embed inline	✓	✓
Accessibility		✓
CSS Styling		✓
Authoring tools		✓
Speed / Memory	✓ ✓	✓

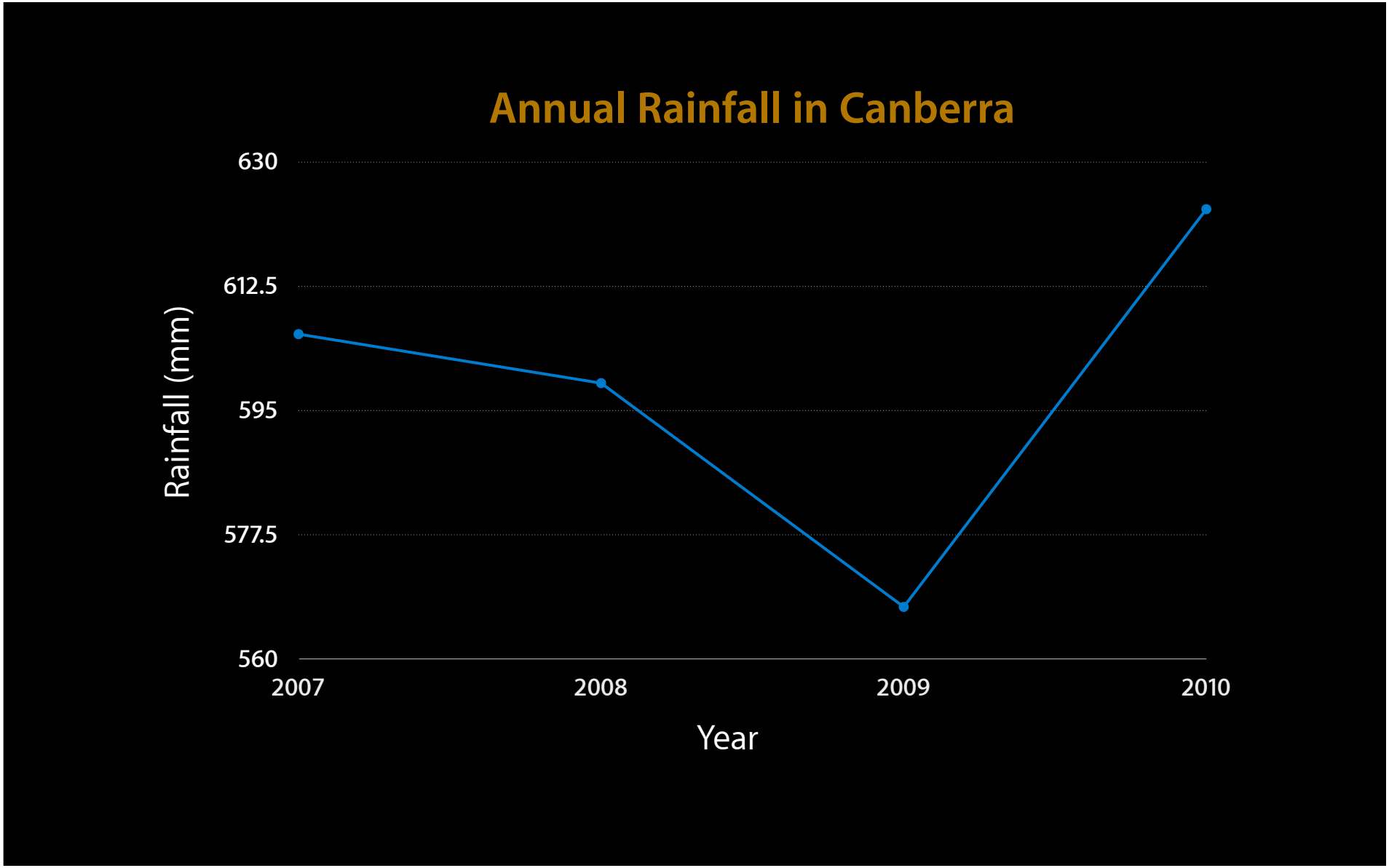
What You Will Learn

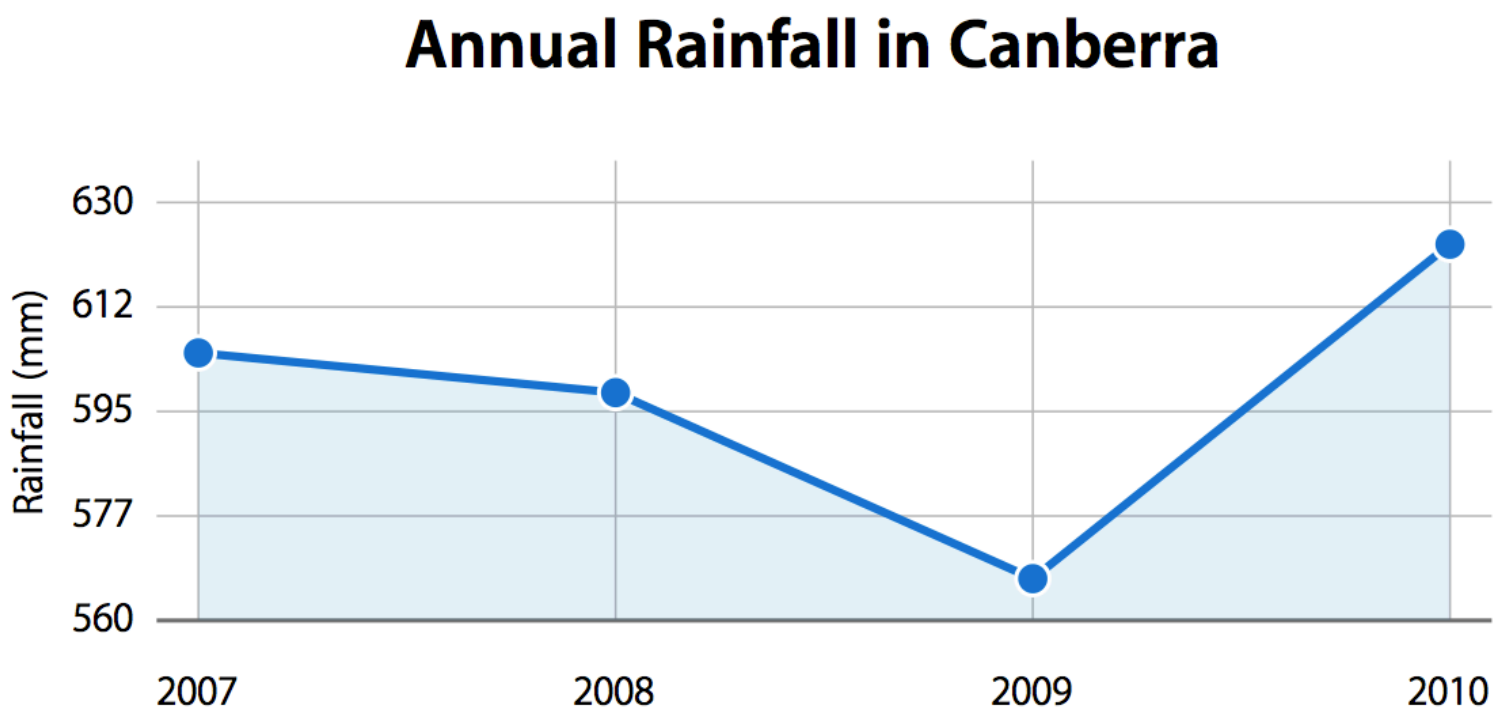
1. Basic Concepts with Canvas and SVG
2. Animation and Interactivity
3. Tips and Special Effects

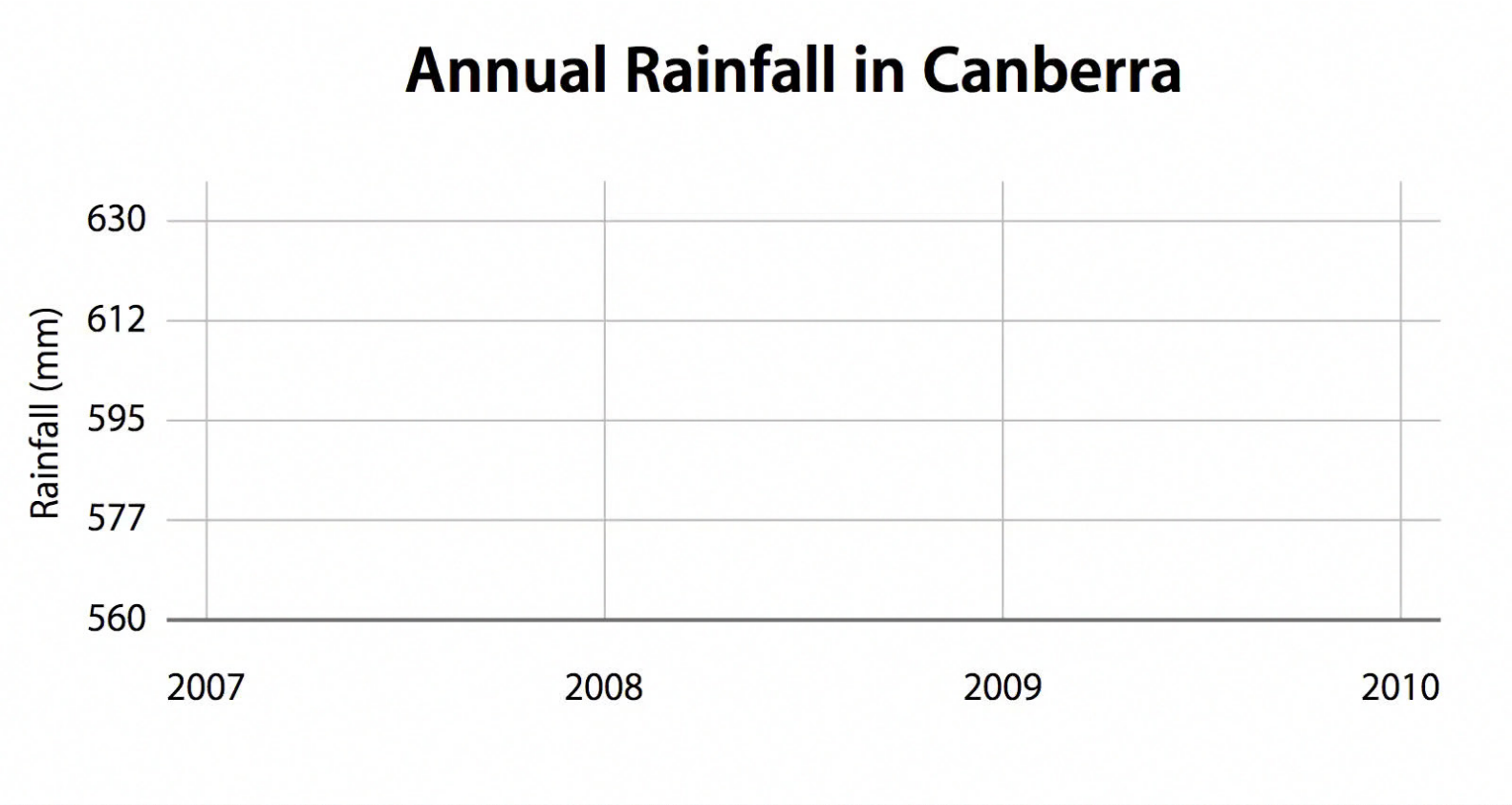
What You Will Learn

1. Basic Concepts with Canvas and SVG
2. Animation and Interactivity
3. Tips and Special Effects

Animation and Interactivity


Annual Rainfall in Canberra
630
612.5
595
577.5
560
Rainfall (mm)
2007
2008
2009
2010
Year


Annual Rainfall in Canberra
Rainfall (mm)
630
612
595
577
560
2007
2008
2009
2010


Annual Rainfall in Canberra
Rainfall (mm)
630
612
595
577
560
2007
2008
2009
2010

Animating with Canvas

Animating with Canvas

```
var DELAY = 1000 / 30; // Aim for 30 fps

function animationStep(fromValue, toValue, startedAt, duration) {

    var elapsed = Date.now() - startedAt;
    var progress = Math.min(elapsed / duration, 1);
    var currentValue = fromValue + (toValue - fromValue) * progress;

    // Draw scene with the new value
    draw(currentValue);

    if (elapsed < duration) {
        setTimeout(function () {
            animationStep(fromValue, toValue, startedAt, duration);
        }, DELAY);
    }

}
```

Animating with Canvas

```
var DELAY = 1000 / 30; // Aim for 30 fps

function animationStep(fromValue, toValue, startedAt, duration) {

    var elapsed = Date.now() - startedAt;
    var progress = Math.min(elapsed / duration, 1);
    var currentValue = fromValue + (toValue - fromValue) * progress;

    // Draw scene with the new value
    draw(currentValue);

    if (elapsed < duration) {
        setTimeout(function () {
            animationStep(fromValue, toValue, startedAt, duration);
        }, DELAY);
    }

}
```

Animating with Canvas

```
var DELAY = 1000 / 30; // Aim for 30 fps

function animationStep(fromValue, toValue, startedAt, duration) {

    var elapsed = Date.now() - startedAt;
    var progress = Math.min(elapsed / duration, 1);
    var currentValue = fromValue + (toValue - fromValue) * progress;

    // Draw scene with the new value
    draw(currentValue);

    if (elapsed < duration) {
        setTimeout(function () {
            animationStep(fromValue, toValue, startedAt, duration);
        }, DELAY);
    }

}
```

Animating with Canvas

```
var DELAY = 1000 / 30; // Aim for 30 fps

function animationStep(fromValue, toValue, startedAt, duration) {

    var elapsed = Date.now() - startedAt;
    var progress = Math.min(elapsed / duration, 1);
    var currentValue = fromValue + (toValue - fromValue) * progress;

    // Draw scene with the new value
    draw(currentValue);

    if (elapsed < duration) {
        setTimeout(function () {
            animationStep(fromValue, toValue, startedAt, duration);
        }, DELAY);
    }

}
```

Animating with Canvas

```
var DELAY = 1000 / 30; // Aim for 30 fps

function animationStep(fromValue, toValue, startedAt, duration) {

    var elapsed = Date.now() - startedAt;
    var progress = Math.min(elapsed / duration, 1);
    var currentValue = fromValue + (toValue - fromValue) * progress;

    // Draw scene with the new value
    draw(currentValue);

    if (elapsed < duration) {
        setTimeout(function () {
            animationStep(fromValue, toValue, startedAt, duration);
        }, DELAY);
    }

}
```

Animating with Canvas

```
var DELAY = 1000 / 30; // Aim for 30 fps

function animationStep(fromValue, toValue, startedAt, duration) {

    var elapsed = Date.now() - startedAt;
    var progress = Math.min(elapsed / duration, 1);
    var currentValue = fromValue + (toValue - fromValue) * progress;

    // Draw scene with the new value
    draw(currentValue);

    if (elapsed < duration) {
        setTimeout(function () {
            animationStep(fromValue, toValue, startedAt, duration);
        }, DELAY);
    }

}
```

Beyond the Basics

- Use an easing function to smooth the animation
- Synchronize multiple animating objects
- Animating complex types like points or matrices

Animating with SVG

Animating with SVG

```
<circle cx="100" cy="100" r="8">

</circle>
```

Animating with SVG

```
<circle cx="100" cy="100" r="8">
  <animate attributeName="cx" from="100" to="200"
           dur="2s"/>
</circle>
```

Animating with SVG

```
<circle cx="100" cy="100" r="8">
  <animate id="slide" attributeName="cx" from="100"
           to="200" dur="2s"/>
  <animate attributeName="cy" from="100" to="80" dur="2s"
           begin="slide.end"/>
</circle>
```

Animating with SVG

```
<circle cx="100" cy="100" r="8">
  <animate id="slide" attributeName="cx" from="100"
           to="200" dur="2s"/>
  <animate attributeName="cy" from="100" to="80" dur="2s"
           begin="slide.end"/>
</circle>
```

Animating with SVG

```
<circle cx="100" cy="100" r="8">
  <animate id="slide" attributeName="cx" from="100"
           to="200" dur="2s"/>
  <animate attributeName="cy" from="100" to="80" dur="2s"
           begin="slide.end + 2s"/>
</circle>
```


Year: 2008
Month: February
Rainfall: 65mm

Interaction with Canvas

```
function isPointInShape(context, x, y) {

    context.beginPath();
    context.moveTo(..., ...);
    ...
    context.closePath();

    return context.isPointInPath(x, y);

}
```

Interaction with Canvas

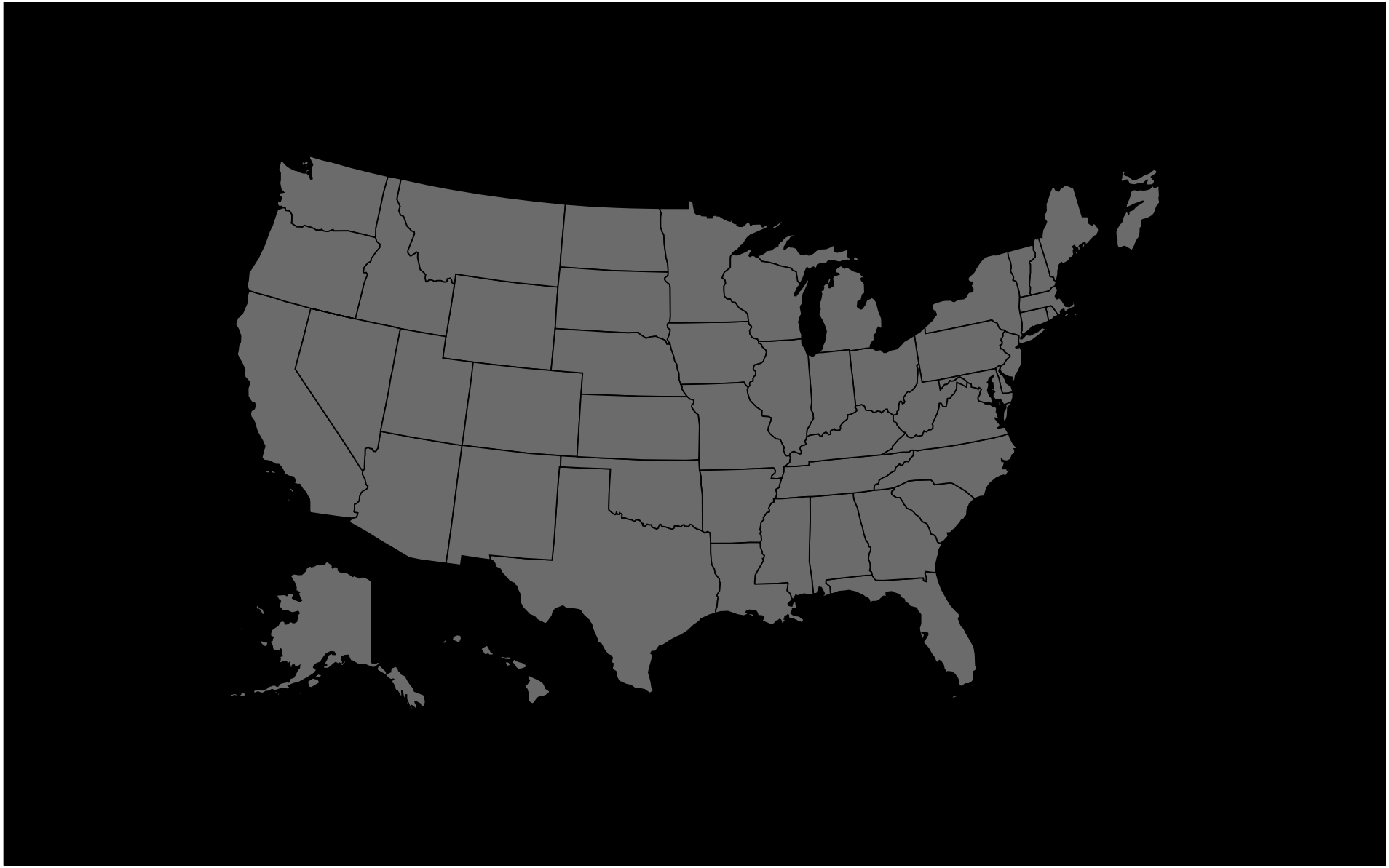
```
function isPointInShape(context, x, y) {

    context.beginPath();
    context.moveTo(..., ...);
    ...
    context.closePath();

    return context.isPointInPath(x, y);

}
```

Interaction with Canvas

```
function isPointInShape(context, x, y) {

    context.beginPath();
    context.moveTo(..., ...);
    ...
    context.closePath();

    return context.isPointInPath(x, y);

}
```

Interaction with Canvas

```
function isPointInShape(context, x, y) {

    context.beginPath();
    context.moveTo(..., ...);
    ...
    context.closePath();

    return context.isPointInPath(x, y);

}
```



Interaction with SVG

```
<polygon id="myShape"
         points="100,100 ..."/>
```

Interaction with SVG

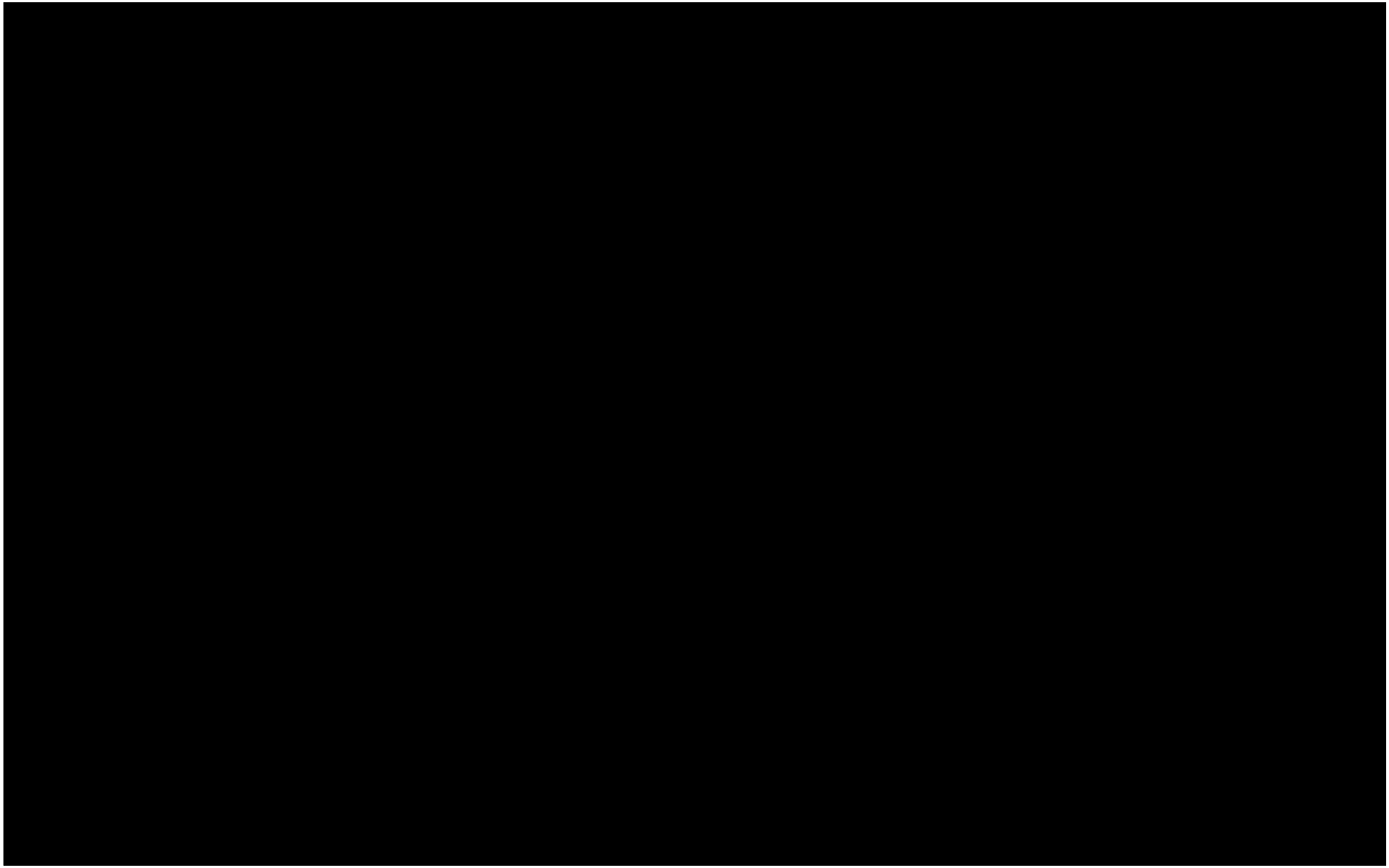
```
<polygon id="myShape"
         points="100,100 ..."/>
```

```
var shape = document.getElementById("myShape");
shape.addEventListener("click",
                       myClickHandler,
                       false);
```

Interaction with SVG

```
<polygon id="myShape"
         points="100,100 ..."/>
```

```
var shape = document.getElementById("myShape");
shape.addEventListener("click",
                       myClickHandler,
                       false);
```



Drawing Images

Drawing Images

```
img = new Image();
img.src = “http://:...”;

// wait for image to load

context.drawImage(img,
                  0, 0);

context.drawImage(img,
                  0, 0,
                  300, 600);

context.drawImage(img,
                  230, 130,
                  250, 250,
                  0, 0,
                  500, 500);
```

Drawing Images

```
img = new Image();
img.src = “http://:...”;

// wait for image to load

context.drawImage(img,
                  0, 0);

context.drawImage(img,
                  0, 0,
                  300, 600);

context.drawImage(img,
                  230, 130,
                  250, 250,
                  0, 0,
                  500, 500);
```

Drawing Images

```
img = new Image();
img.src = “http://:...”;

// wait for image to load

context.drawImage(img,
                  0, 0);

context.drawImage(img,
                  0, 0,
                  300, 600);

context.drawImage(img,
                  230, 130,
                  250, 250,
                  0, 0,
                  500, 500);
```

Drawing Images

```
img = new Image();
img.src = “http://:...”;

// wait for image to load

context.drawImage(img,
                  0, 0);

context.drawImage(img,
                  0, 0,
                  300, 600);

context.drawImage(img,
                  230, 130,
                  250, 250,
                  0, 0,
                  500, 500);
```

Drawing Images

```
img = new Image();
img.src = “http://:...”;

// wait for image to load

context.drawImage(img,
                  0, 0);

context.drawImage(img,
                  0, 0,
                  300, 600);

context.drawImage(img,
                  230, 130,
                  250, 250,
                  0, 0,
                  500, 500);
```

THE INCIDENT

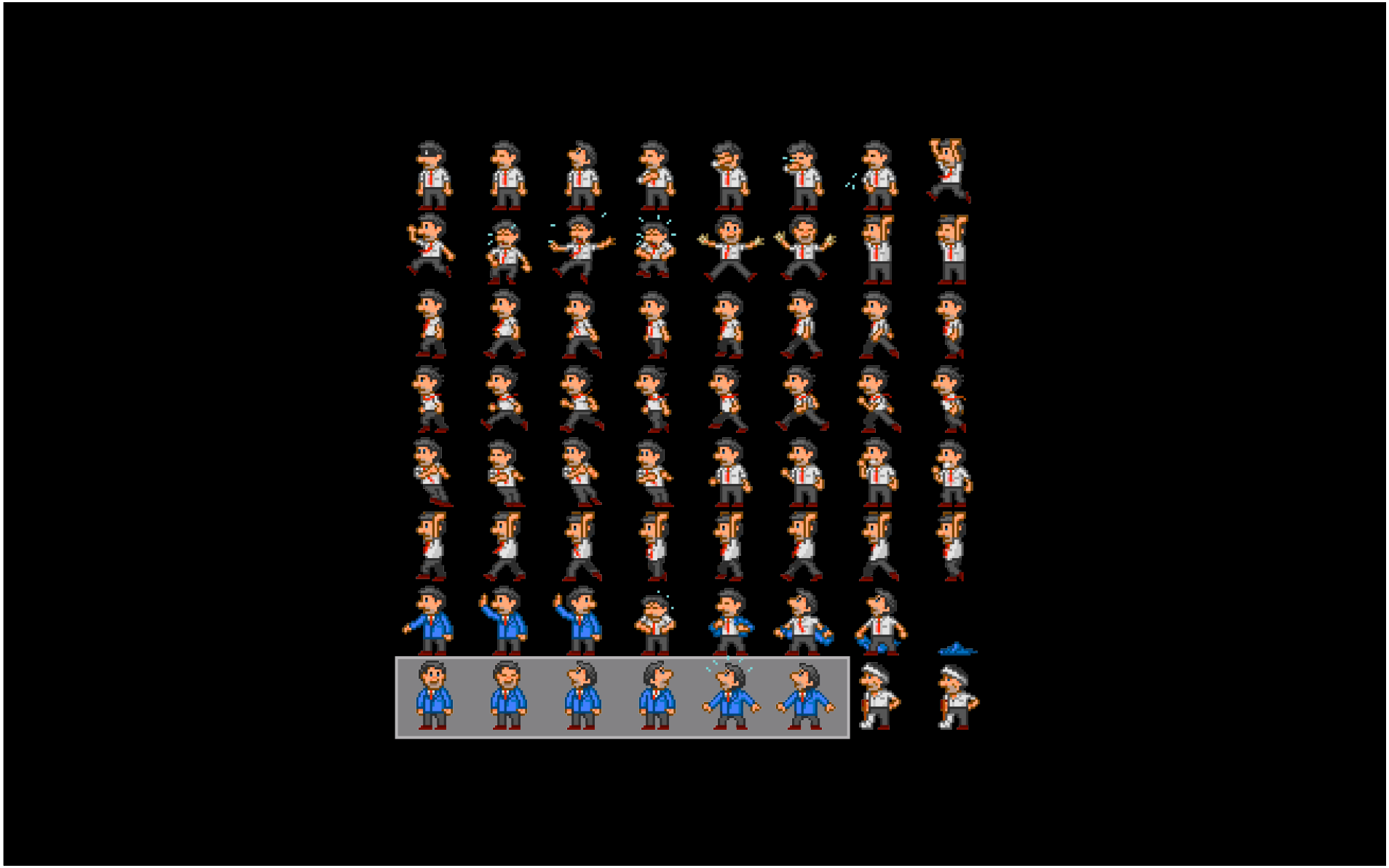
THE INCIDENT

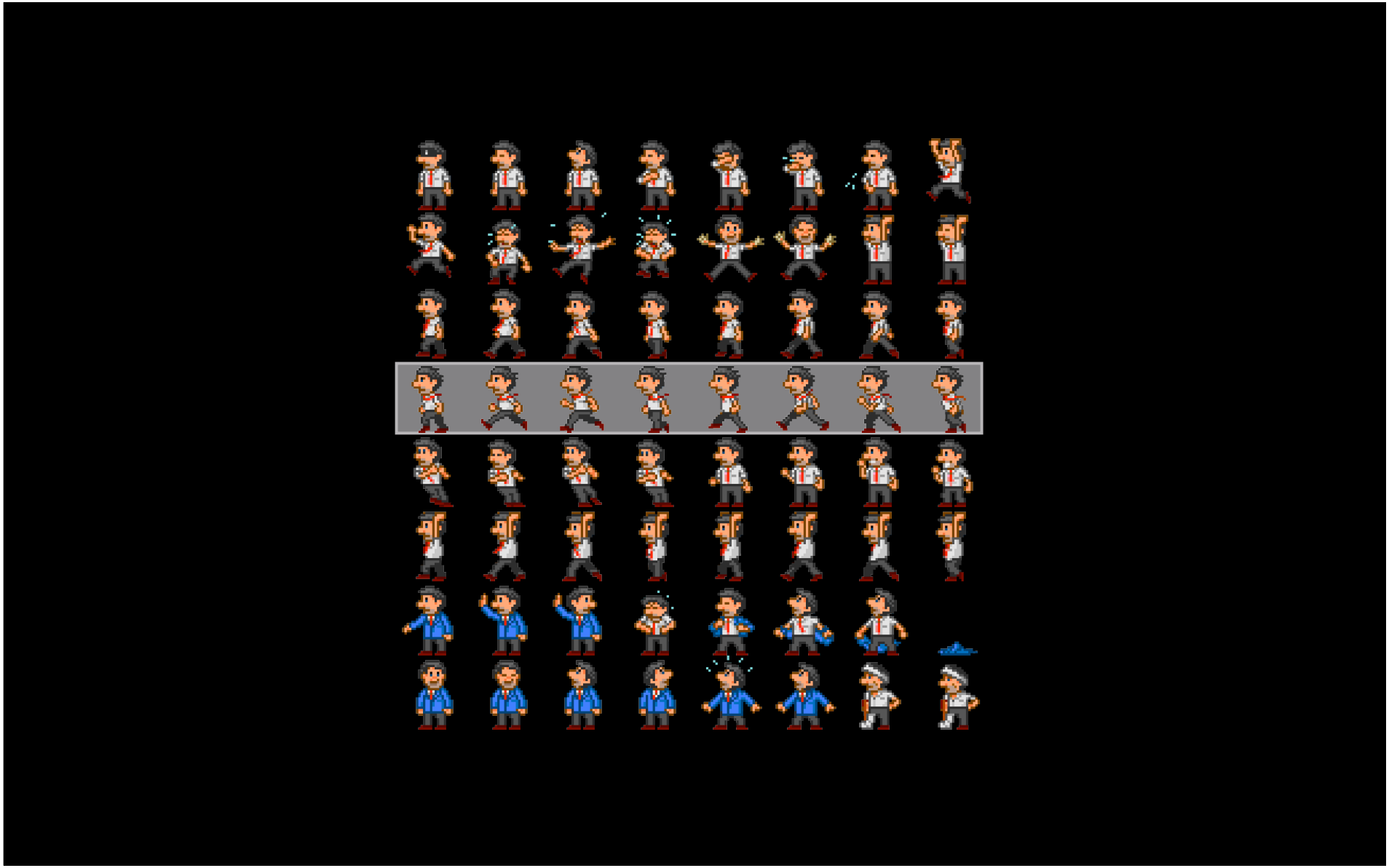
Available on the App Store

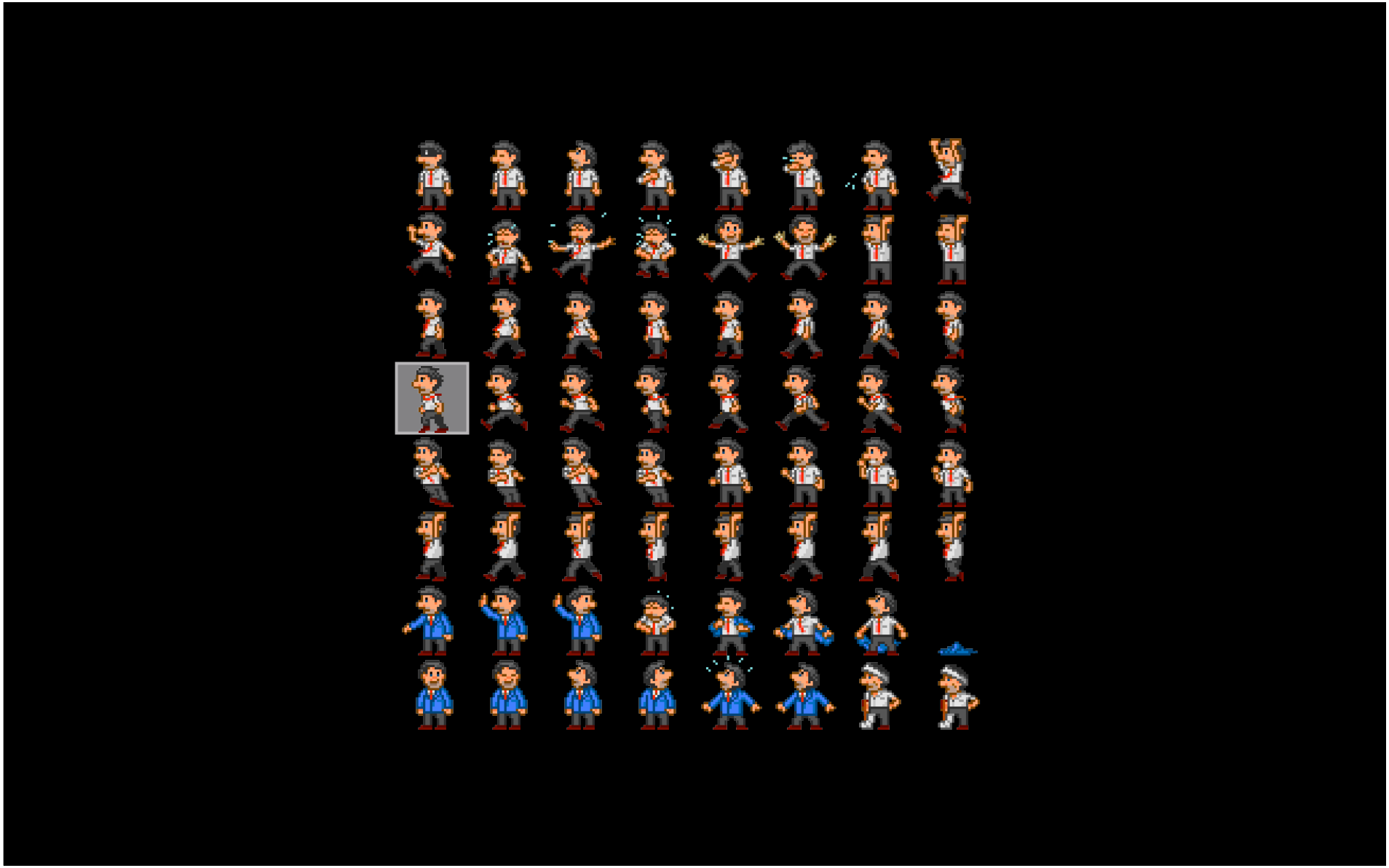
Available on the Mac App Store

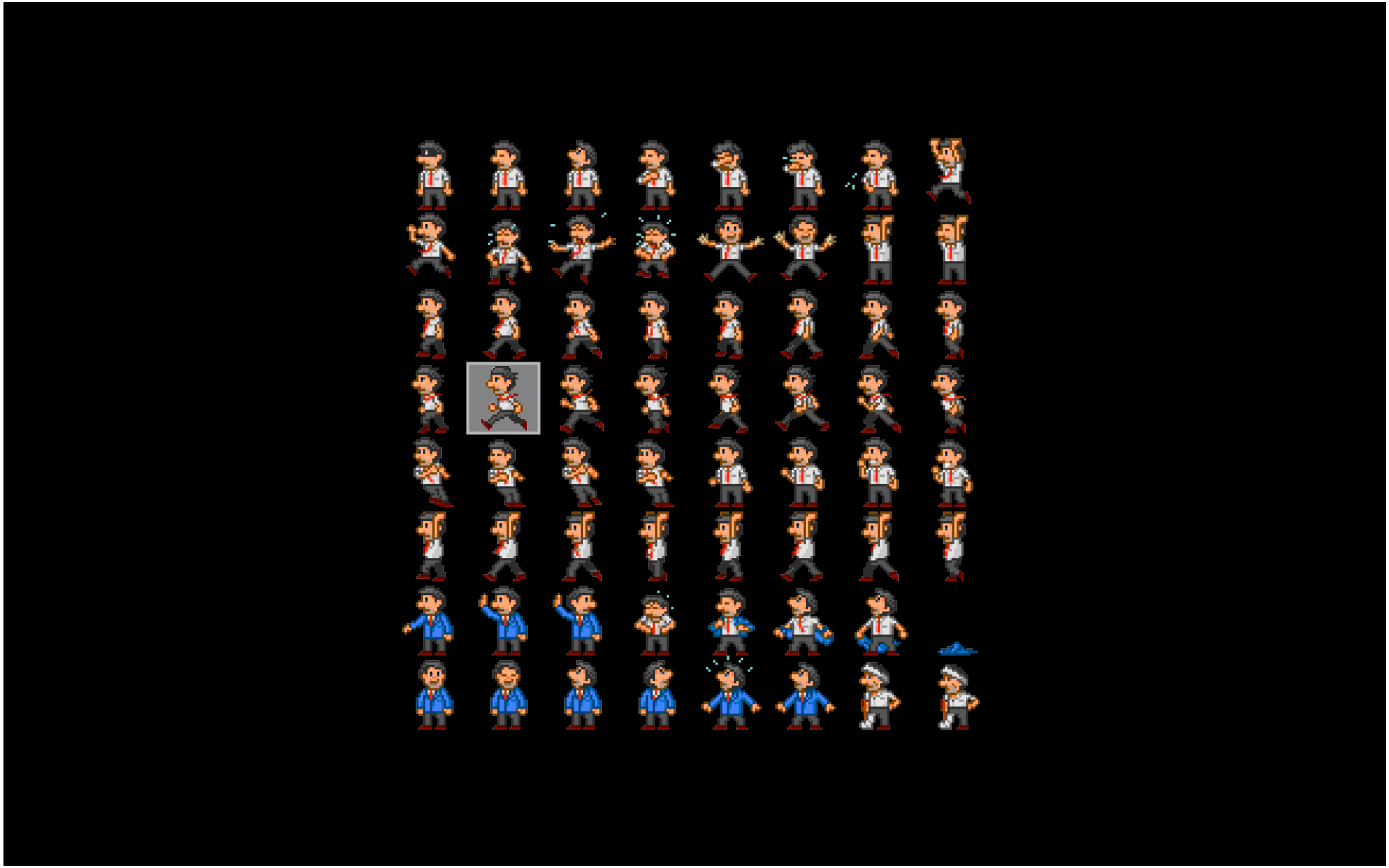
THE INCIDENT

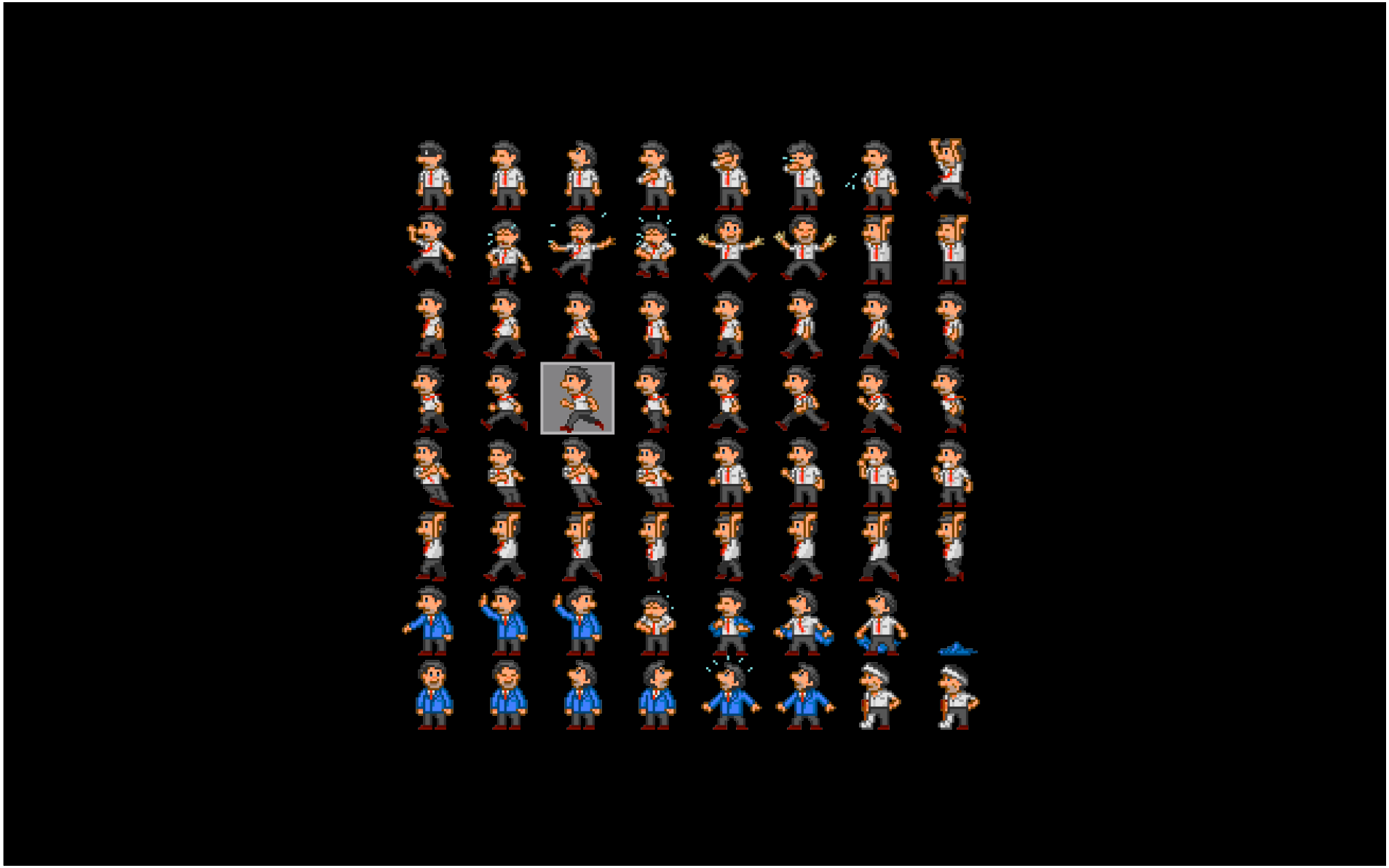
“The Incident” Sprite Sheet

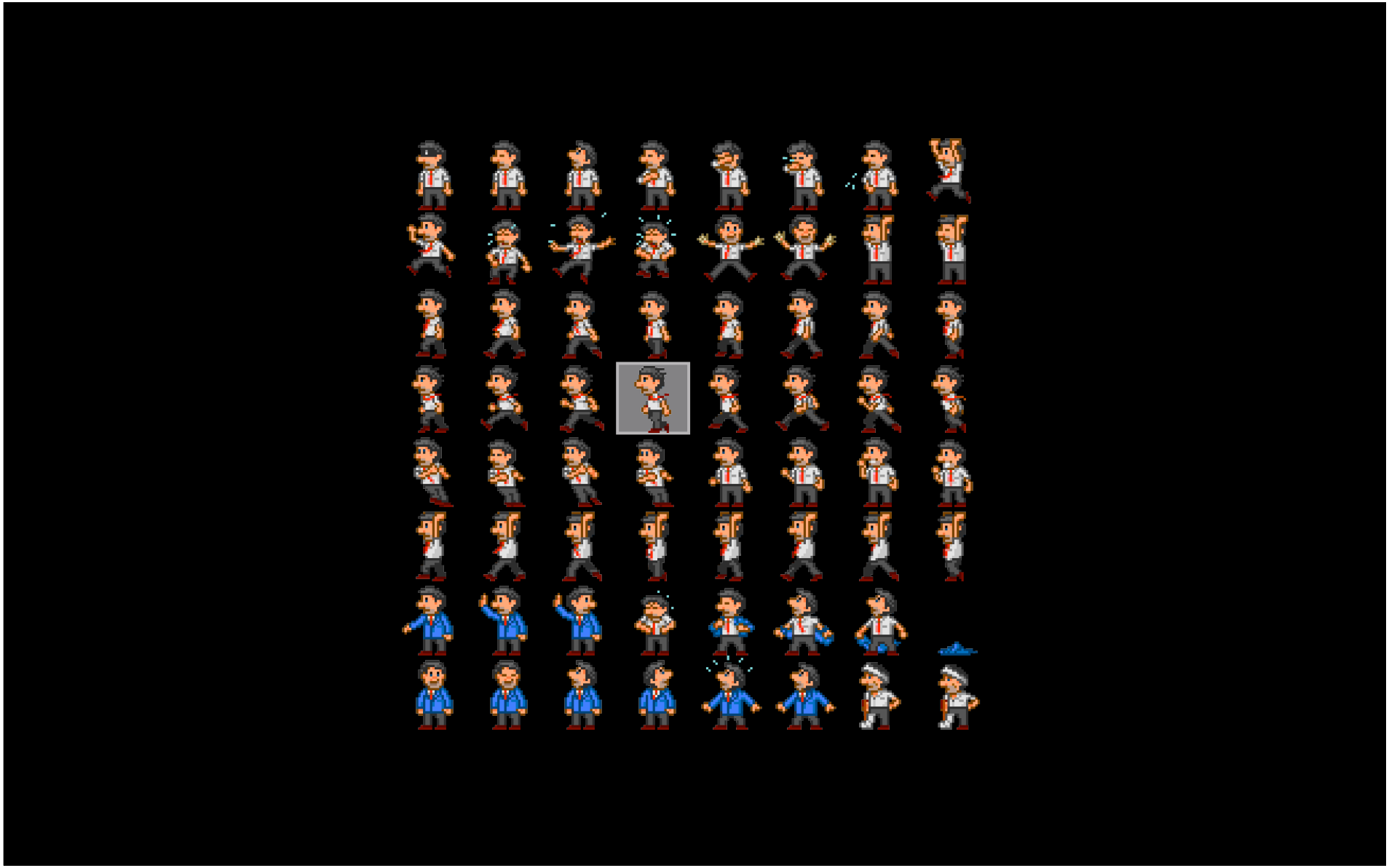


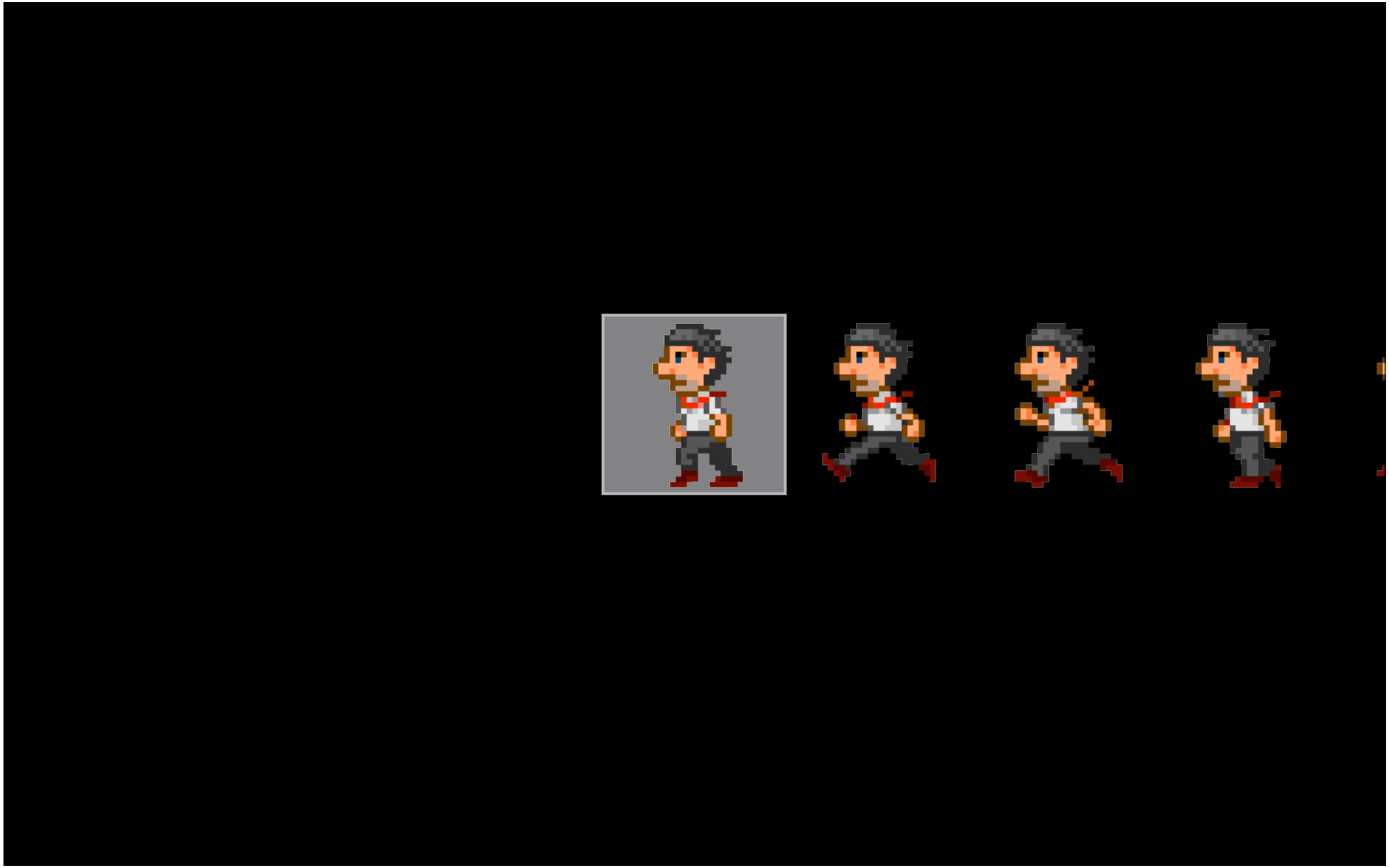


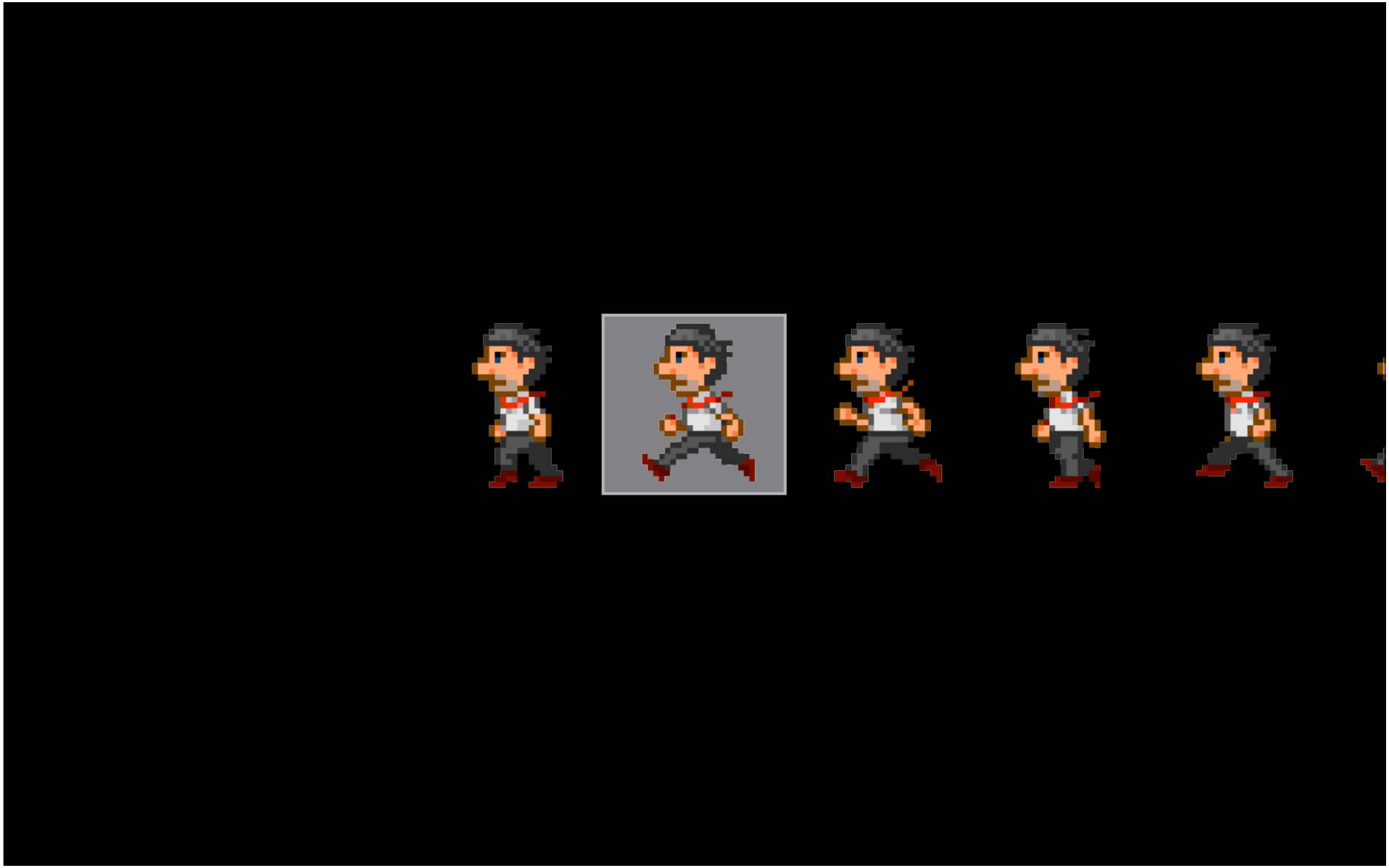


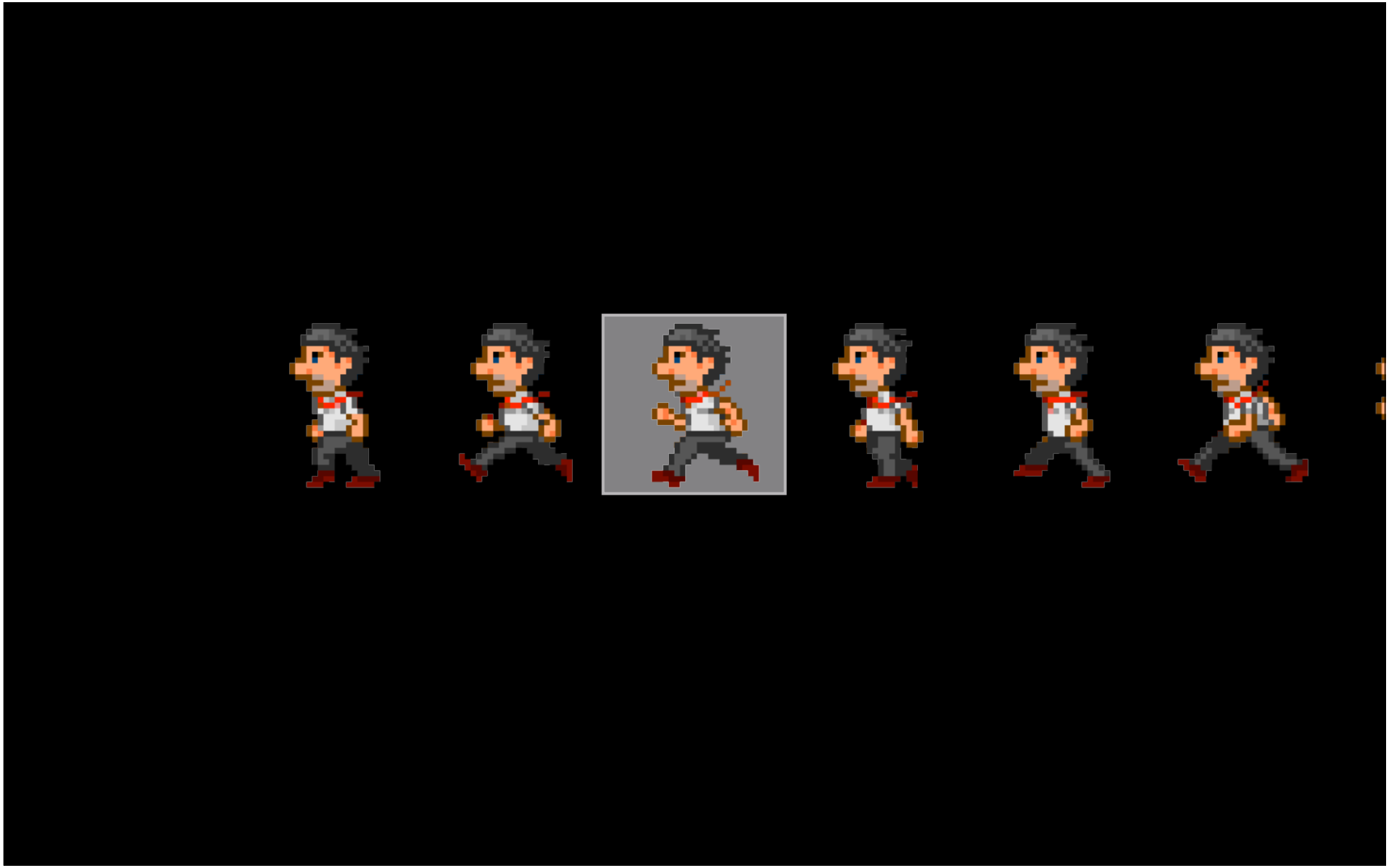


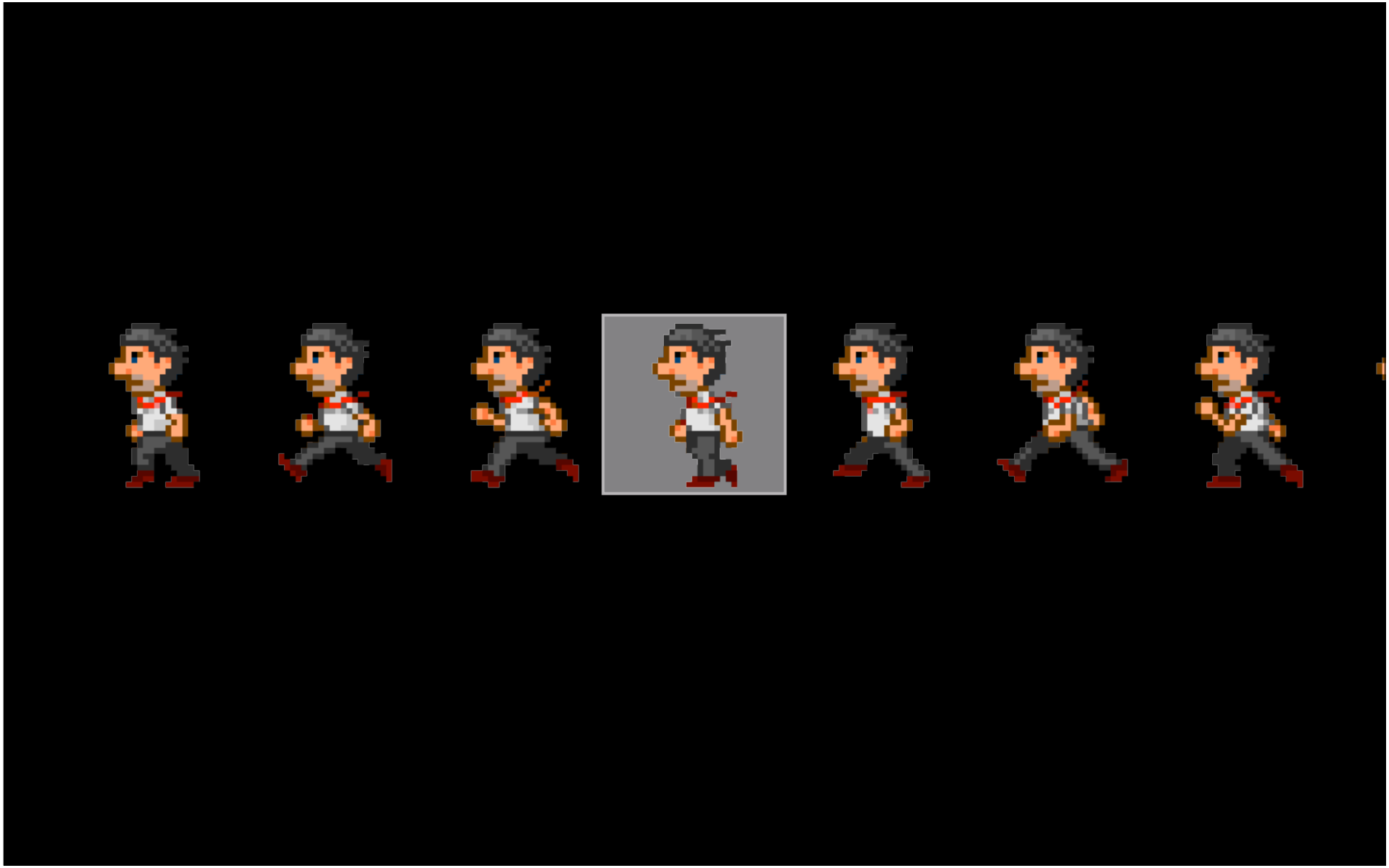


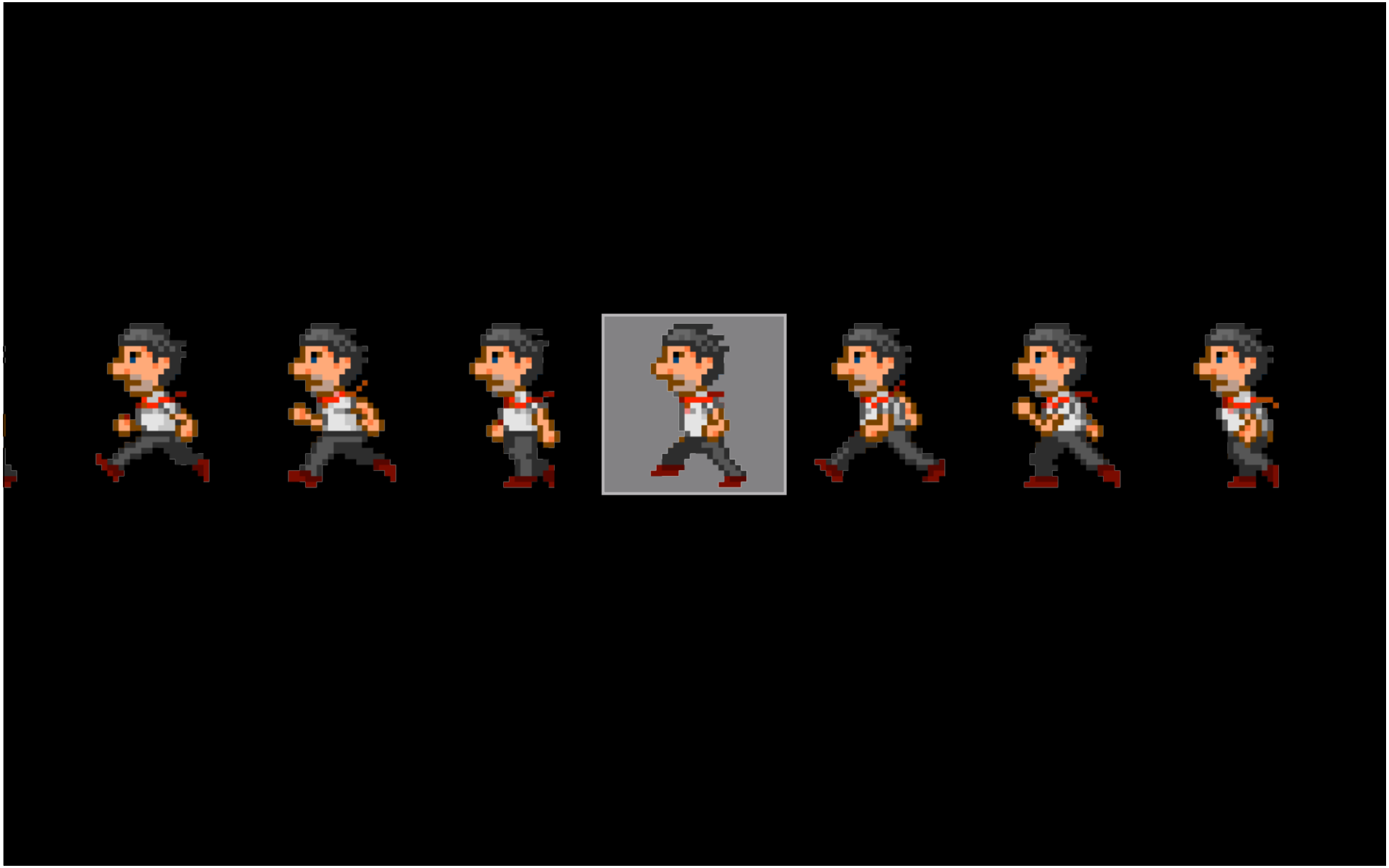


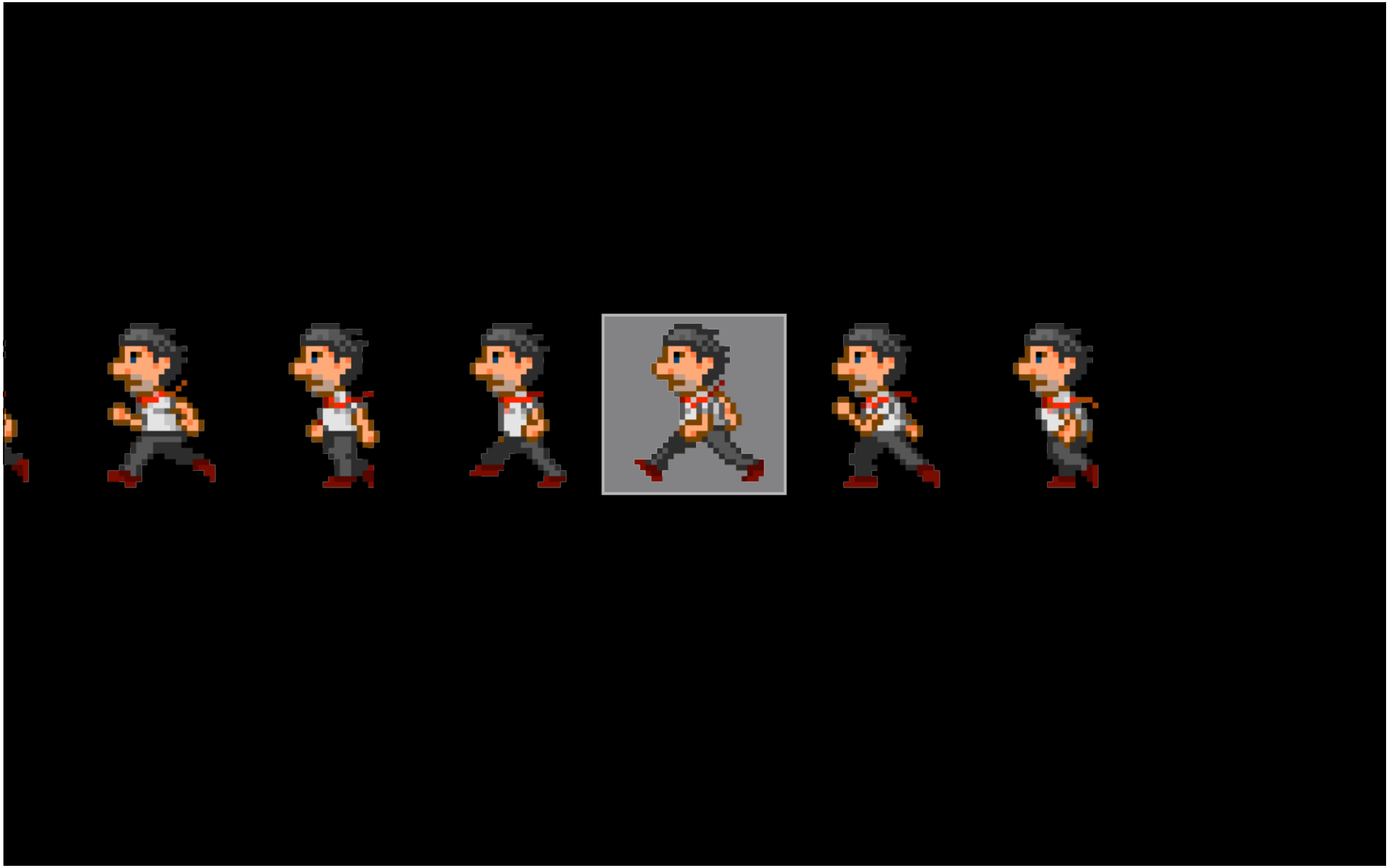












Demo

What You Will Learn

1. Basic Concepts with Canvas and SVG
2. Animation and Interactivity
3. Tips and Special Effects

What You Will Learn

1. Basic Concepts with Canvas and SVG
2. Animation and Interactivity
3. Tips and Special Effects

Tips and Special Effects

I want to manipulate pixels

Accessing Pixels

```
var pixels = context.getImageData(x, y, width, height);
```

```
pixels.data = [ R G B A R G B A R G B A … ];
```

Manipulating Pixels

```
var input = context.getImageData(0, 0, width, height);
var output = context.createImageData(width, height);

for (var i=0; i < (width * height); i++) {
  // for every pixel
  output.data[i * 4] = // some operation
  output.data[i * 4 + 1] = // some operation
  output.data[i * 4 + 2] = // some operation
  output.data[i * 4 + 3] = // some operation
}

context.drawImage(output, 0, 0, width, height);
```

Manipulating Pixels

```
var input = context.getImageData(0, 0, width, height);
var output = context.createImageData(width, height);

for (var i=0; i < (width * height); i++) {
  // for every pixel
  output.data[i * 4] = // some operation
  output.data[i * 4 + 1] = // some operation
  output.data[i * 4 + 2] = // some operation
  output.data[i * 4 + 3] = // some operation
}

context.drawImage(output, 0, 0, width, height);
```

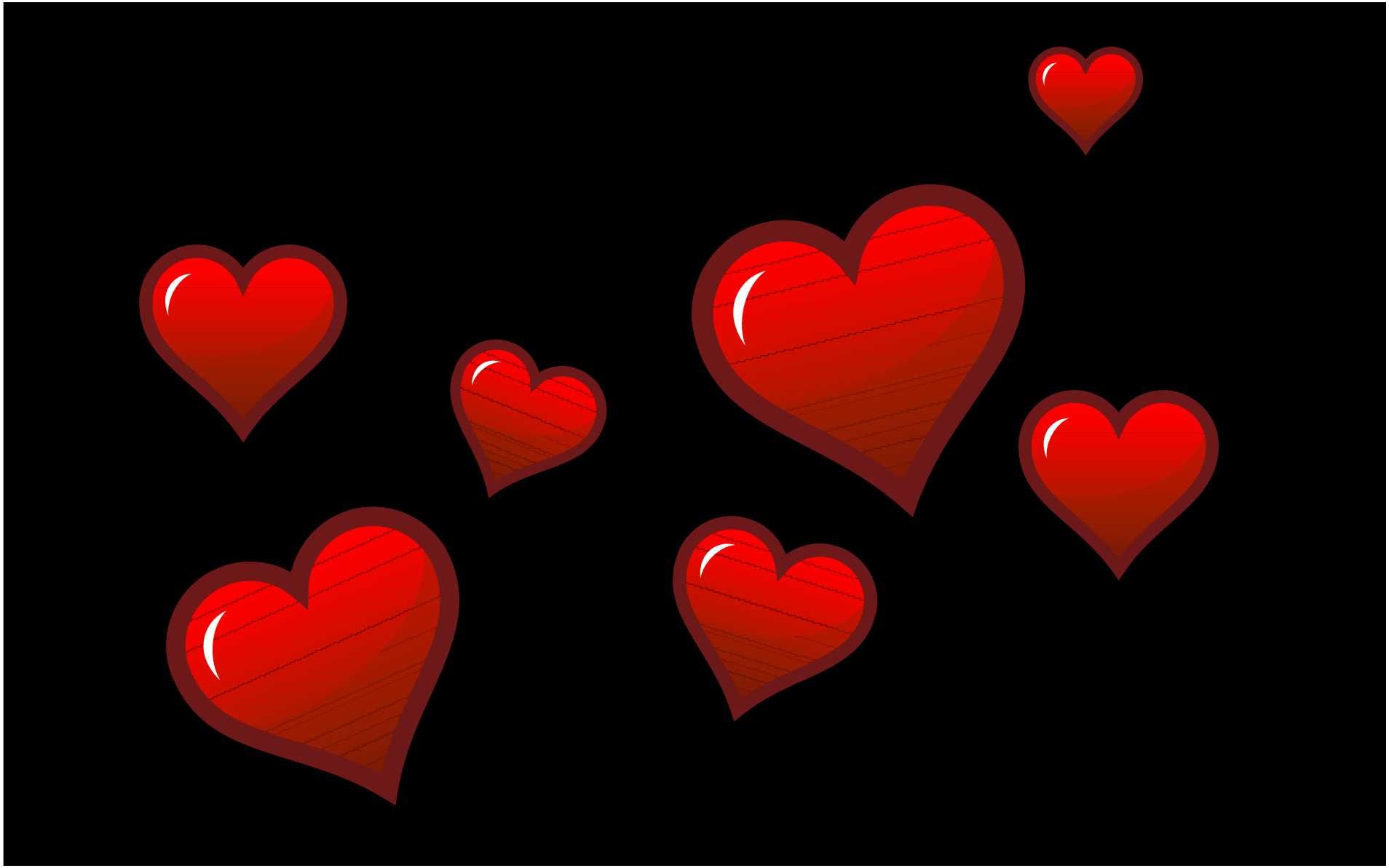
Manipulating Pixels

```
var input = context.getImageData(0, 0, width, height);
var output = context.createImageData(width, height);

for (var i=0; i < (width * height); i++) {
  // for every pixel
  output.data[i * 4] = // some operation
  output.data[i * 4 + 1] = // some operation
  output.data[i * 4 + 2] = // some operation
  output.data[i * 4 + 3] = // some operation
}

context.drawImage(output, 0, 0, width, height);
```

Manipulating Pixels

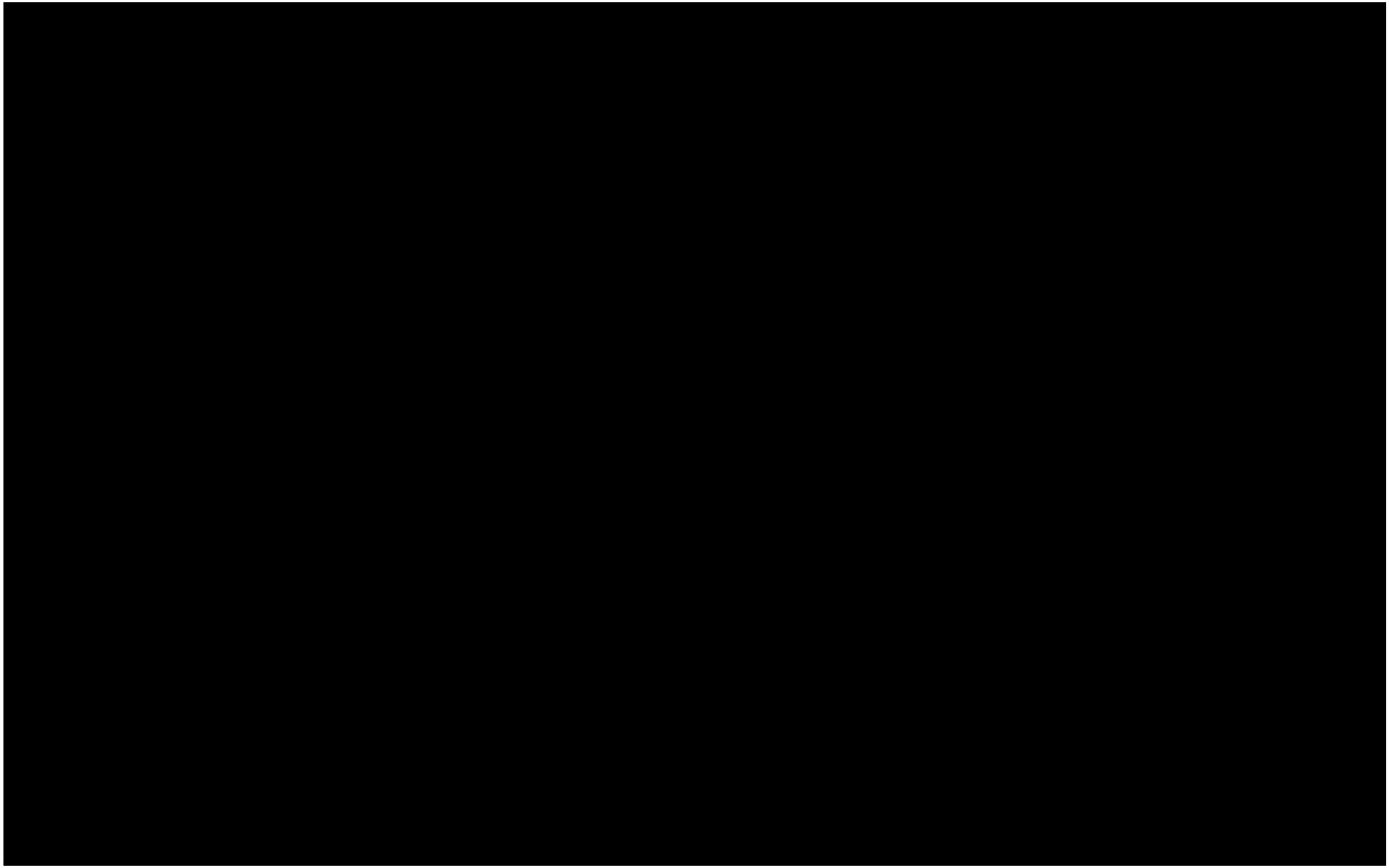
```
var input = context.getImageData(0, 0, width, height);
var output = context.createImageData(width, height);

for (var i=0; i < (width * height); i++) {
  // for every pixel
  output.data[i * 4] = // some operation
  output.data[i * 4 + 1] = // some operation
  output.data[i * 4 + 2] = // some operation
  output.data[i * 4 + 3] = // some operation
}

context.drawImage(output, 0, 0, width, height);
```

I want to draw lots of repeated shapes as fast as possible



Offscreen Buffers

```
var offscreen = document.createElement(“canvas”);
offscreen.width = 100;
offscreen.height = 100;

var offscreenContext = offscreen.getContext(“2d”);
drawHeart(offscreenContext, 100, 100);
```



Offscreen Buffers

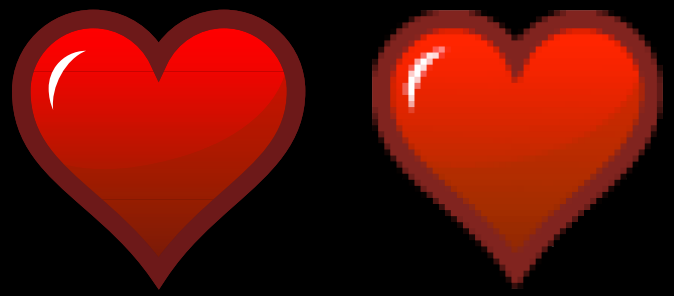
```
var offscreen = document.createElement(“canvas”);
offscreen.width = 100;
offscreen.height = 100;

var offscreenContext = offscreen.getContext(“2d”);
drawHeart(offscreenContext, 100, 100);
```

Offscreen Buffers

```
var offscreen = document.createElement(“canvas”);
offscreen.width = 100;
offscreen.height = 100;

var offscreenContext = offscreen.getContext(“2d”);
drawHeart(offscreenContext, 100, 100);
```

Offscreen Buffers

```
var offscreen = document.createElement("canvas");
offscreen.width = 100;
offscreen.height = 100;

var offscreenContext = offscreen.getContext("2d");
drawHeart(offscreenContext, 100, 100);


mainContext.drawImage(offscreen, x, y);
```



I want to save my canvas image, or let the user save it

Getting a URL for the Image Data

```
var imageAsString = canvas.toDataURL(“image/png”);
```

Getting a URL for the Image Data

Now what?

- You have a string that you could…
  - place in local storage
  - send back to a server
  - use as the src attribute for a new `<img>` element

Demo

What You Will Learn

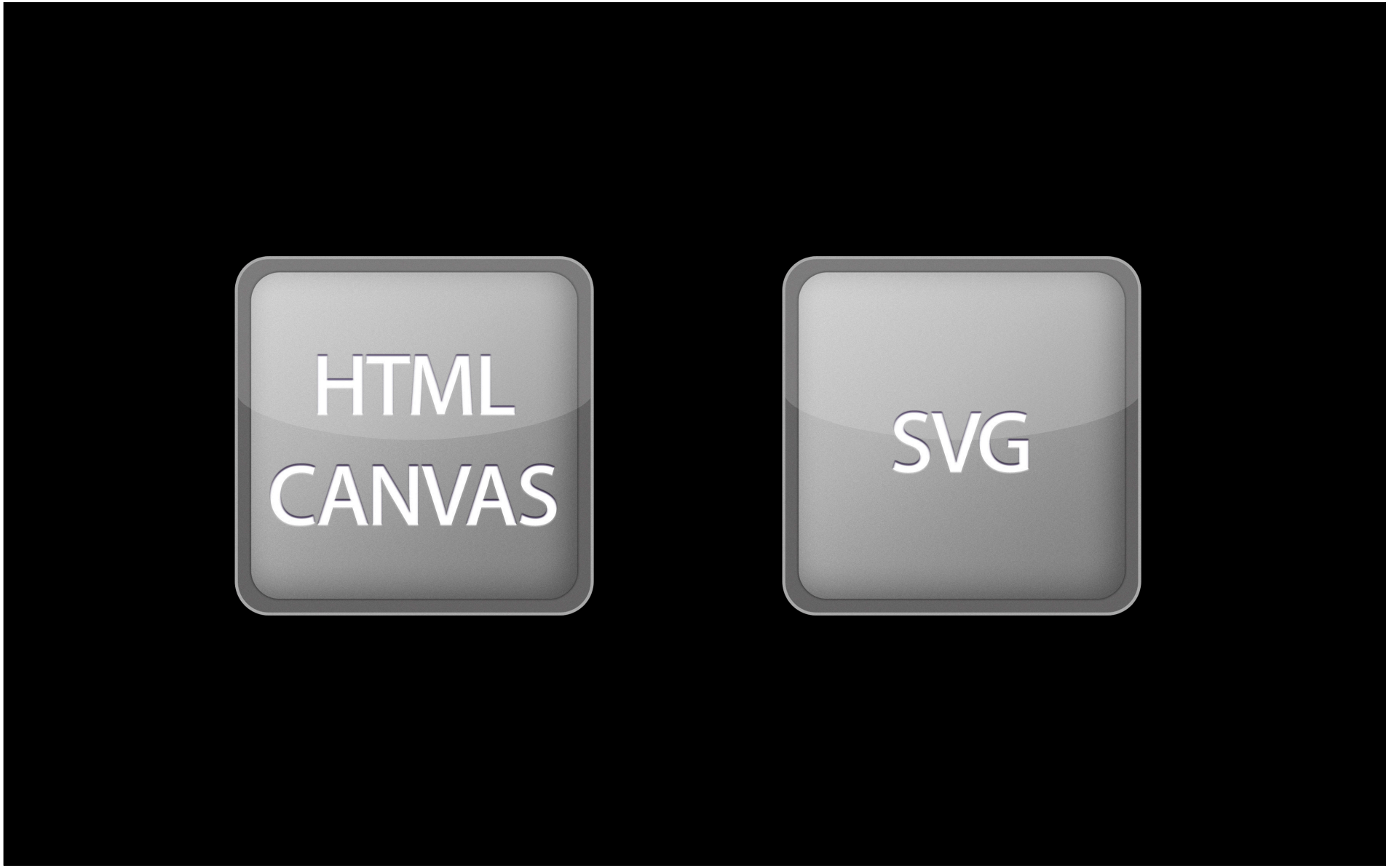
1. Basic Concepts with Canvas and SVG
2. Animation and Interactivity
3. Tips and Special Effects

What You Have Learnt

1. Basic Concepts with Canvas and SVG
2. Animation and Interactivity
3. Tips and Special Effects

Wrapping Up

And The Winner Is…


HTML
CANVAS
SVG

Related Sessions

iBooks: Create Beautiful Books with HTML5, CSS3 and EPUB	Nob Hill Wednesday 2:00 PM
What's New in CSS Effects and Animations	Marina Wednesday 4:30 PM

Labs

Safari on iOS Open Lab	Internet and Web Lab Thursday 9:00 AM
CSS Effects Lab	Internet and Web Lab Thursday 9:00 AM

More Information

Vicki Murley
Safari Technologies Evangelist
vicki@apple.com

Other Resources
http://developer.apple.com/safari
http://www.w3.org/TR/html5
http://www.w3.org/Graphics/SVG

Apple Developer Forums
http://devforums.apple.com

Q&A


WWDC2011