



Velkommen

Tervetuloa

Bienvenue

Welkom

2012 Esri 中国开发者大会

欢迎

欢迎您

Willkomm

Bienvenido

أَهْلًا وَسَهْلًا

ようこそ



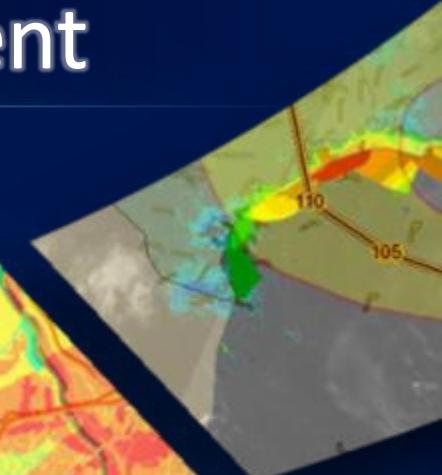
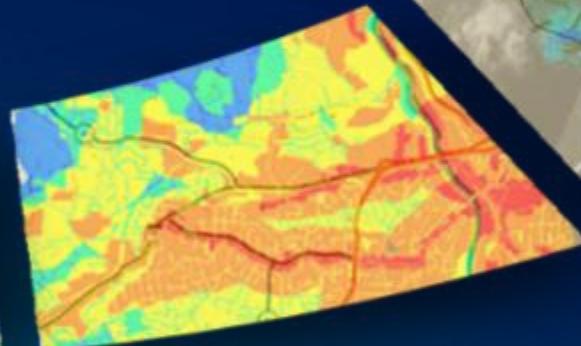


2012

Esri中国开发者大会
2012 Esri China Developer Summit

HTML5 Canvas 2D Drawing Software Development

2012-5



Abstract

HTML5标准提供了Web浏览器中2D绘图的一个通用画布——Canvas，对以往基于Web插件的绘图方式产生了根本性的变化。它解决了基于Web插件的绘图软件（如SVG、VML、Flash或其他各种Plugin/ActiveX控件）存在的无法跨平台、跨浏览器，非通用性，安全性差等诸多问题。

HTML5 Canvas目前已经成为现代浏览器绘图的事实上的标准，许多浏览器（如Chrome、FireFox、Safari、Opera）都提供了很好的支持。

本报告着眼于研究基于HTML5 Canvas的2D GIS图形绘制系统。以JavaScript为实现语言，采用OOP的封装方式，把原来基于插件的实现方式过渡到以Canvas为标准的现代浏览器平台上。并对实现过程中需要注意的细节问题提出了一些解决方法或建议。

Agenda

- Web Drawing History
- HTML5 Canvas Drawing Introduction
- Web-based GIS Drawing Software Design
- Puzzle Issues
- JavaScript Programming
- Demo: A Workflow Editor

Q&A

Web Drawing History – Plugins

SVG, VML, Flash, Silverlight...

Browser

```
<OBJECT ID="Canvas"  
       CLASSID="CLSID:BC3D7FCC-C1AE-  
           ????-????-431457A1173C">  
</OBJECT>
```

JavaScript

Plugin

Native API

OS API

Web Drawing History – Canvas / WebGL

HTML5 Browser

```
<canvas id="_canvasMain"  
       width="300"  
       height="100">  
</canvas>
```

Canvas/WebGL

HTML5 Canvas ... - Platforms

<http://html5test.com>

Browser	Score
Maxthon 3.3.7	437
Chrome 18	400
Firefox 12	345
Opera 11.60	338
Safari 5.1	317
IE9	138

HTML5 Canvas ... - Benefits

- Browser built-in capability
- Sandbox-based safety
- Hardware acceleration
- Sub-pixel rendering
- Immediate mode
- JavaScript portal
- Cross OS and browser compatible

HTML5 Canvas ... - Drawbacks

- Canvas: bitmapped area of the screen
- Not a retained mode: drawing objects persistence left to caller
- JavaScript-driven: complex and slow
- Hard to debug
- Biz Logic Exposure
- Memory-predatory

HTML5 Canvas Introduction - Functionalities

- Various *shapes*: point, line, curve, rectangle, fill, path
- *Texts*
- *Images*
- *Colors, Gradients, Shadows*
- *Rotations*
- *Alpha Transparencies*
- *Pixel Manipulations*
- *Matrix Transformations*
- *Events (Keyboard and Mouse)*

HTML5 Canvas ... - Drawings



HTML5 Canvas ... - Performance

Performance Improving:

<http://www.html5rocks.com/en/tutorials/canvas/performance/>

- Batch canvas calls together
- Avoid unnecessary canvas state changes
- Render screen differences only, not the whole new state
- Use multiple layered canvases for complex scenes
- Know various ways to clear the canvas
- Most mobile canvas implementations are slow
- Never use canvas' transformation (added by me)

Web-based GIS Drawing Software Design - Rules

- OOP & Objects Model

Canvas

Layers

Layer

Layer

Shapes

Shape

Shape

Points

Point

Point

Renders

Render

Render

Symbols

Symbol

Symbol

Web-based GIS Drawing ... - Interfaces

Canvas Object

Attributes:

- width
- Height
- bkgndColor
- bkgndImage
- layers
- widgets
- dataBound
-

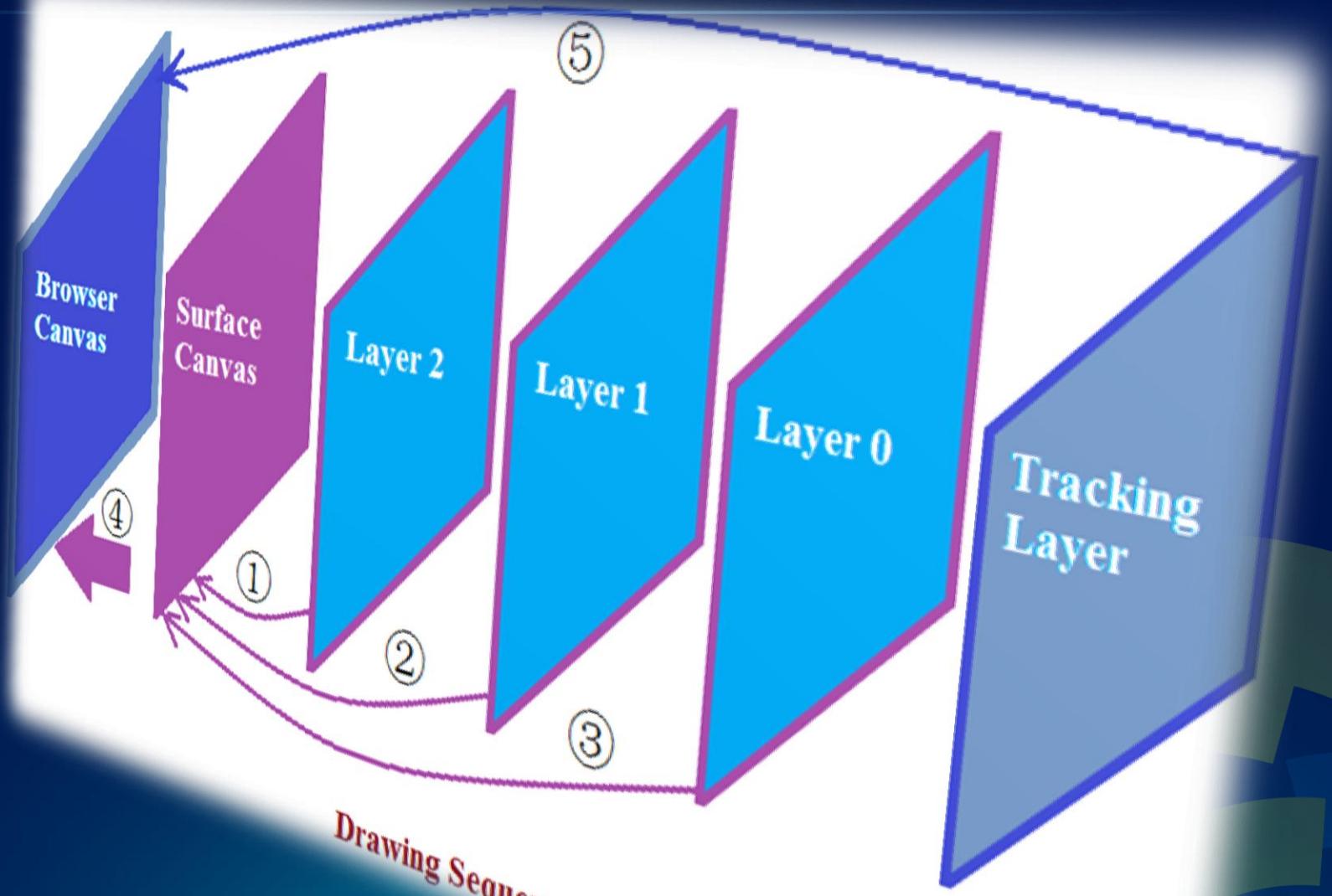
Methods:

- resizeMap
- refresh
- toMapPoint
- toViewPoint
- zoom
- moveLayer
- updateBound
- hitTest
-

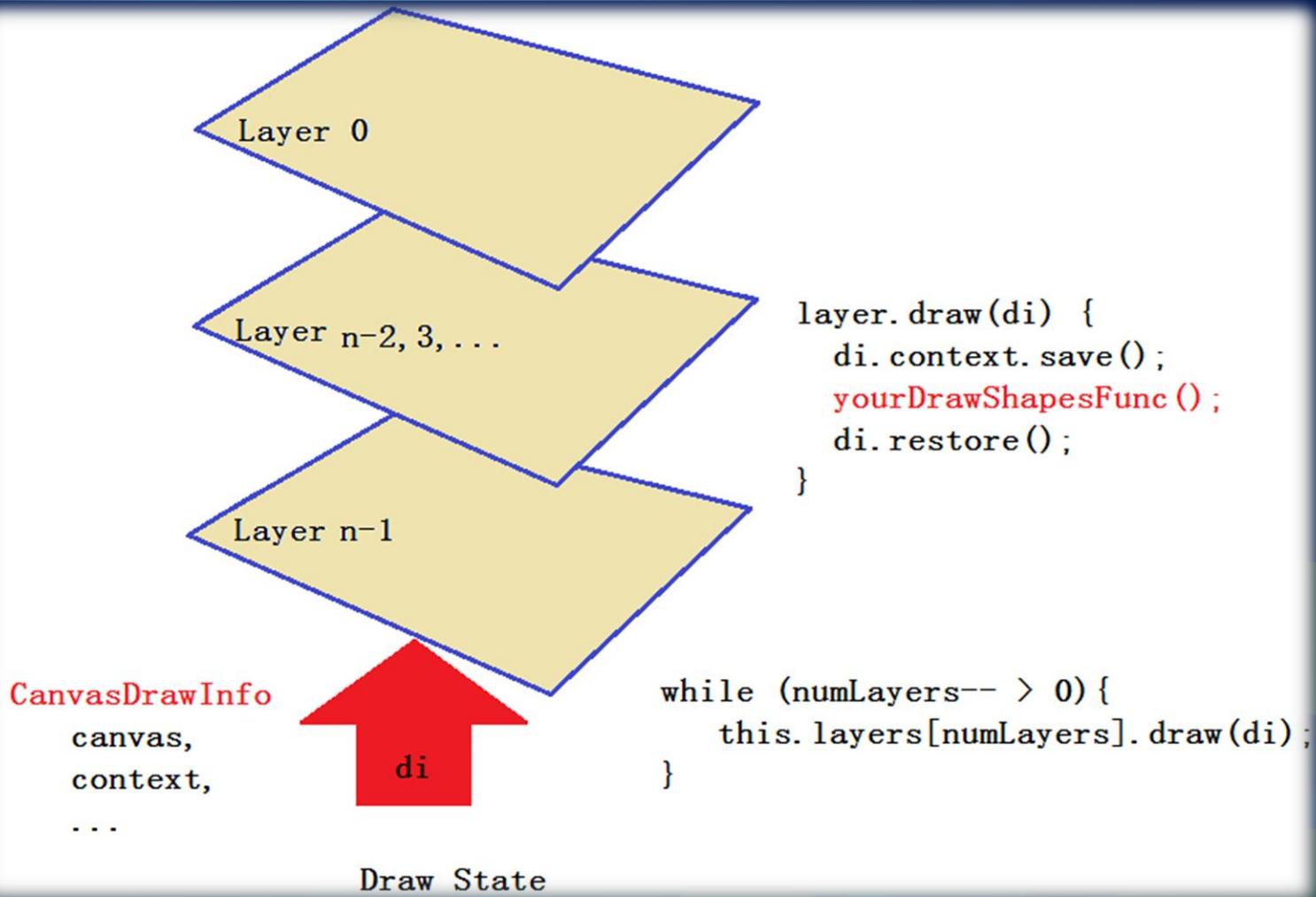
Events:

- onKeyPressed
- onMouseClick
- onMouseDblClick
- onMouseMove
- onMouseWheel
- onMouseDown
- onMouseUp
-

Web-based GIS Drawing



Web-based GIS Drawing

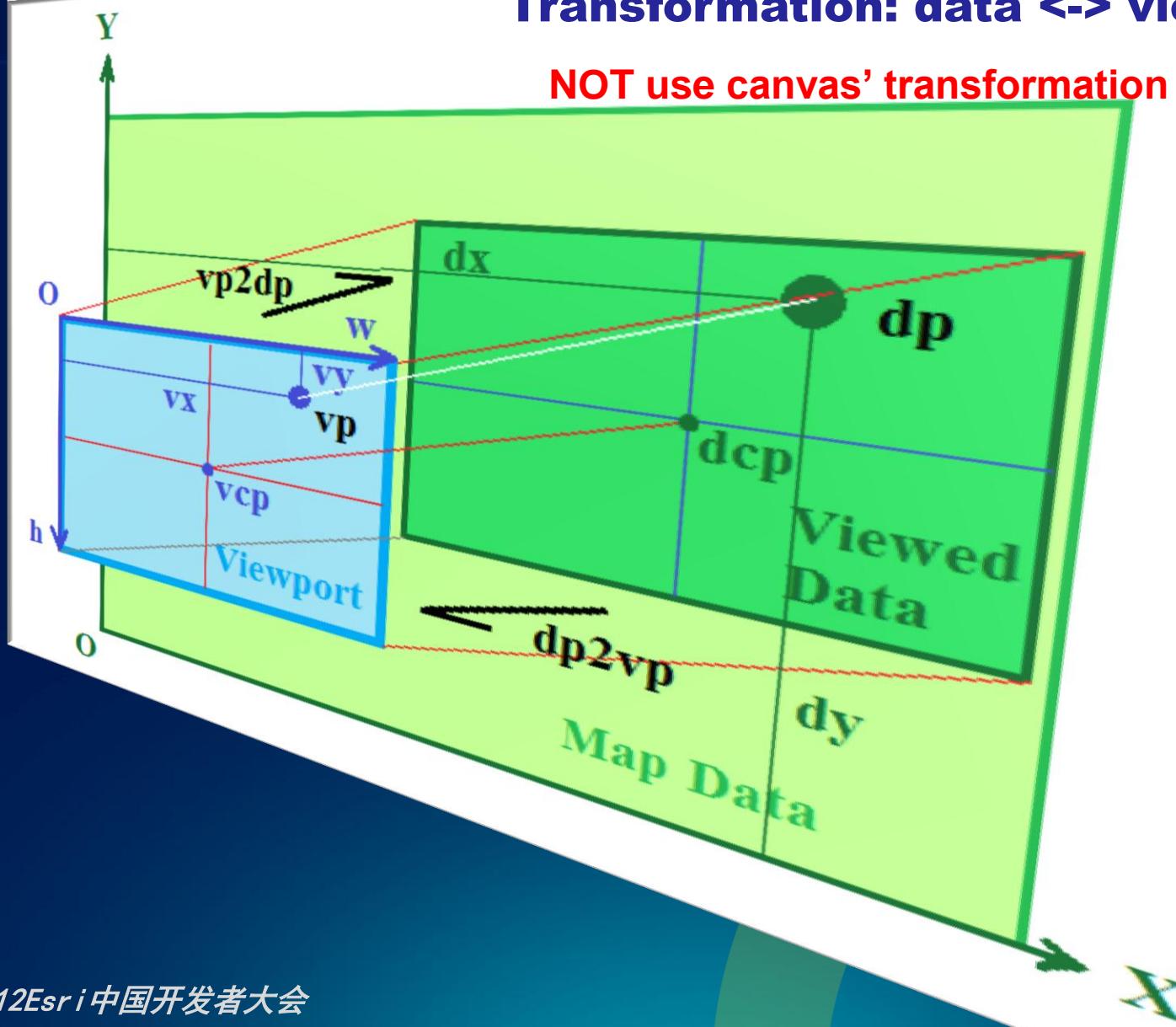


Puzzle Issues

- Transformation: data \leftrightarrow view
- HitTest: where am I
- Anti-Alias disable / Bresenham

Transformation: data <-> view

NOT use canvas' transformation

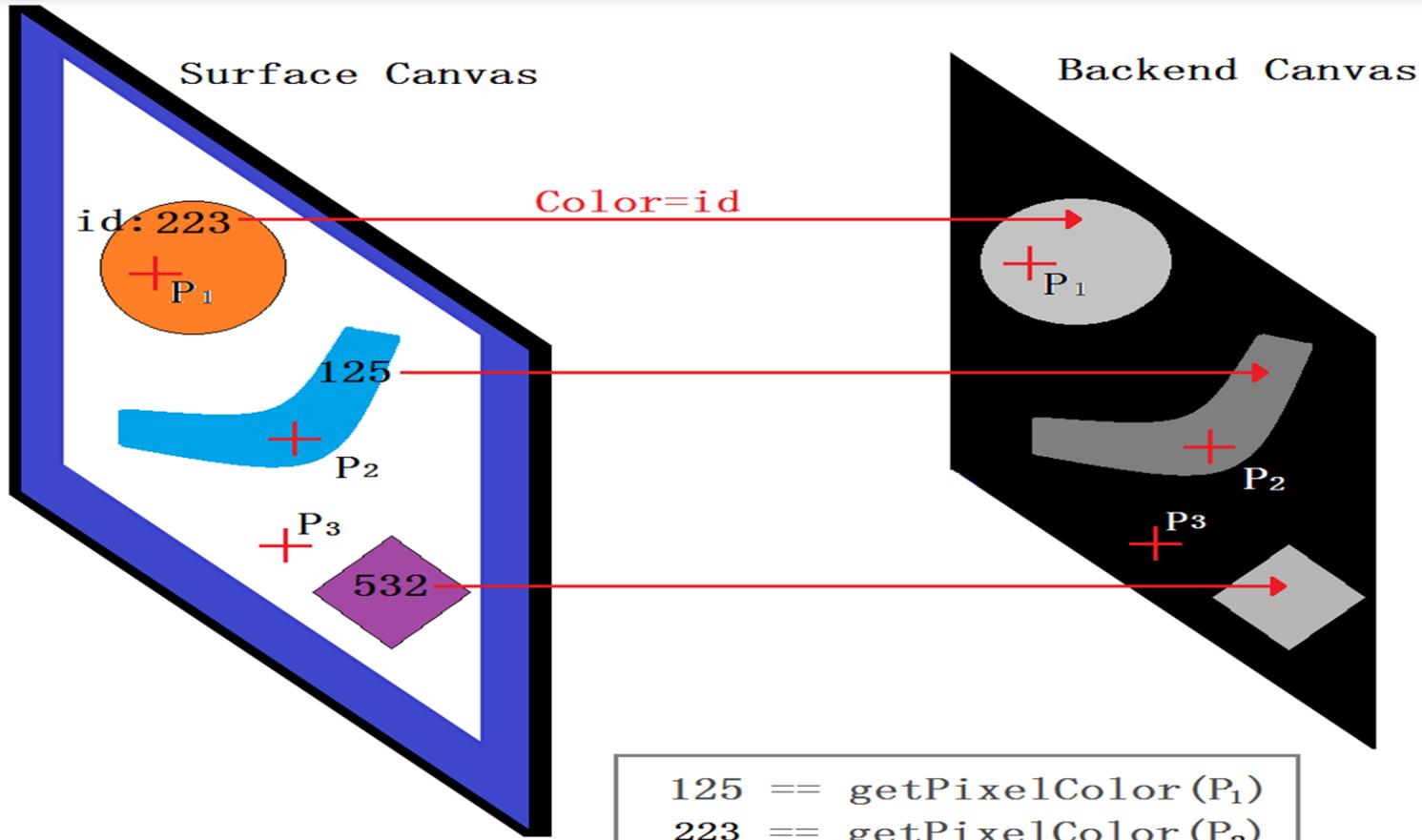


HitTest: where am I

- Which shape hit on?
- Which Image hit on?
- Which Text hit on?
- HitTest on mouse move



HitTest: PixelColor=ShapeID



Pixel Manipulation

```
// Get imageData to direct pixel operation
var imageObj = backendContext.getImageData( 0, 0,
canvas.width, canvas.height);
var imageData = imageObj.data;

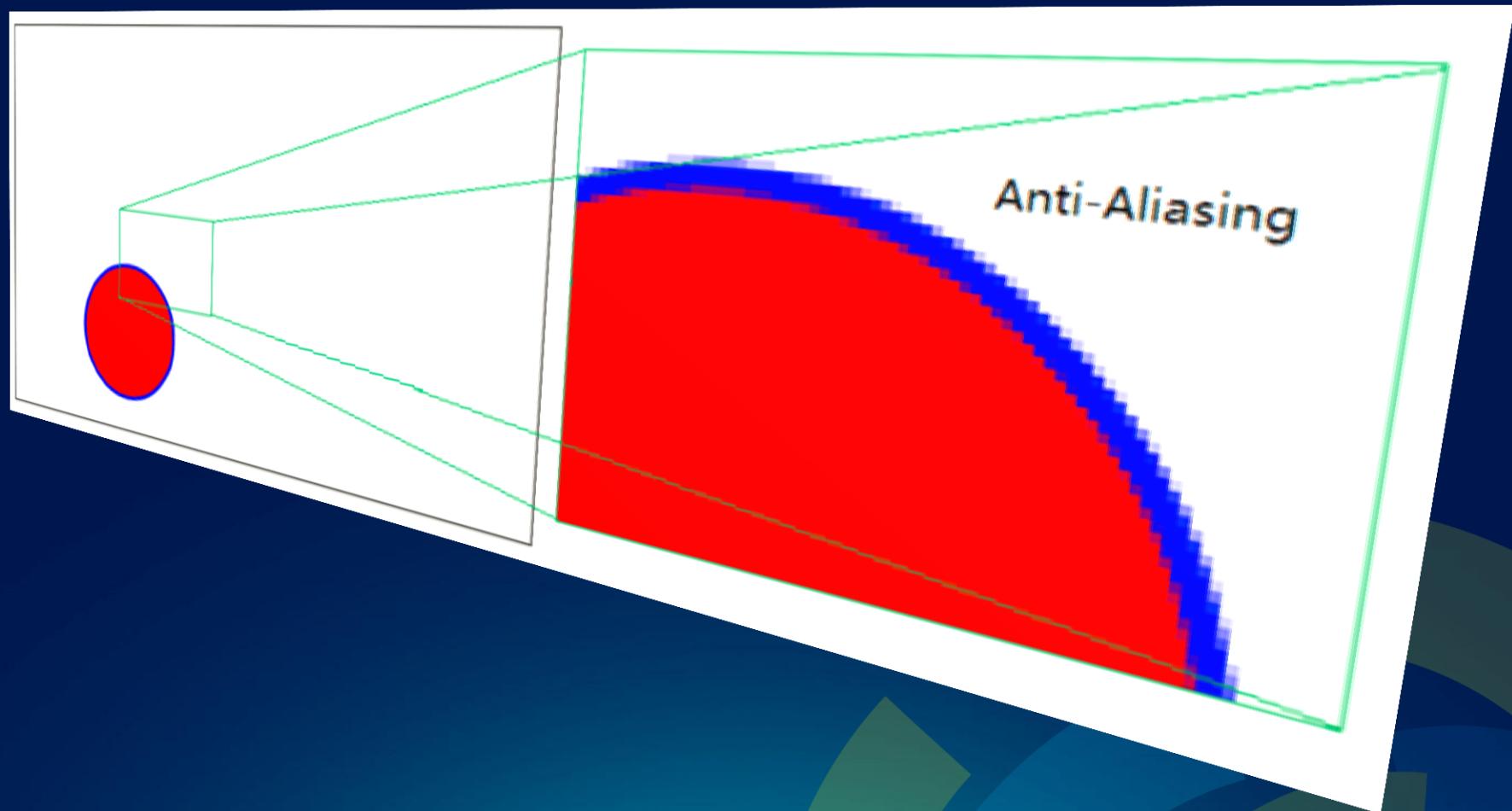
// Use Bresenham to draw pixel on imageData directly
// NOTE that we use shape id as its pixel color
function Bresenham_drawLine(imageData, width, height, x1, y1,
                           x2, y2, r, g, b);
function Bresenham_drawCircle(imageData, width, height, x0, y0,
                           r, r, g, b);
...
...

// Set back imageData to context
backendContext.putImageData(imageData, 0, 0);
```

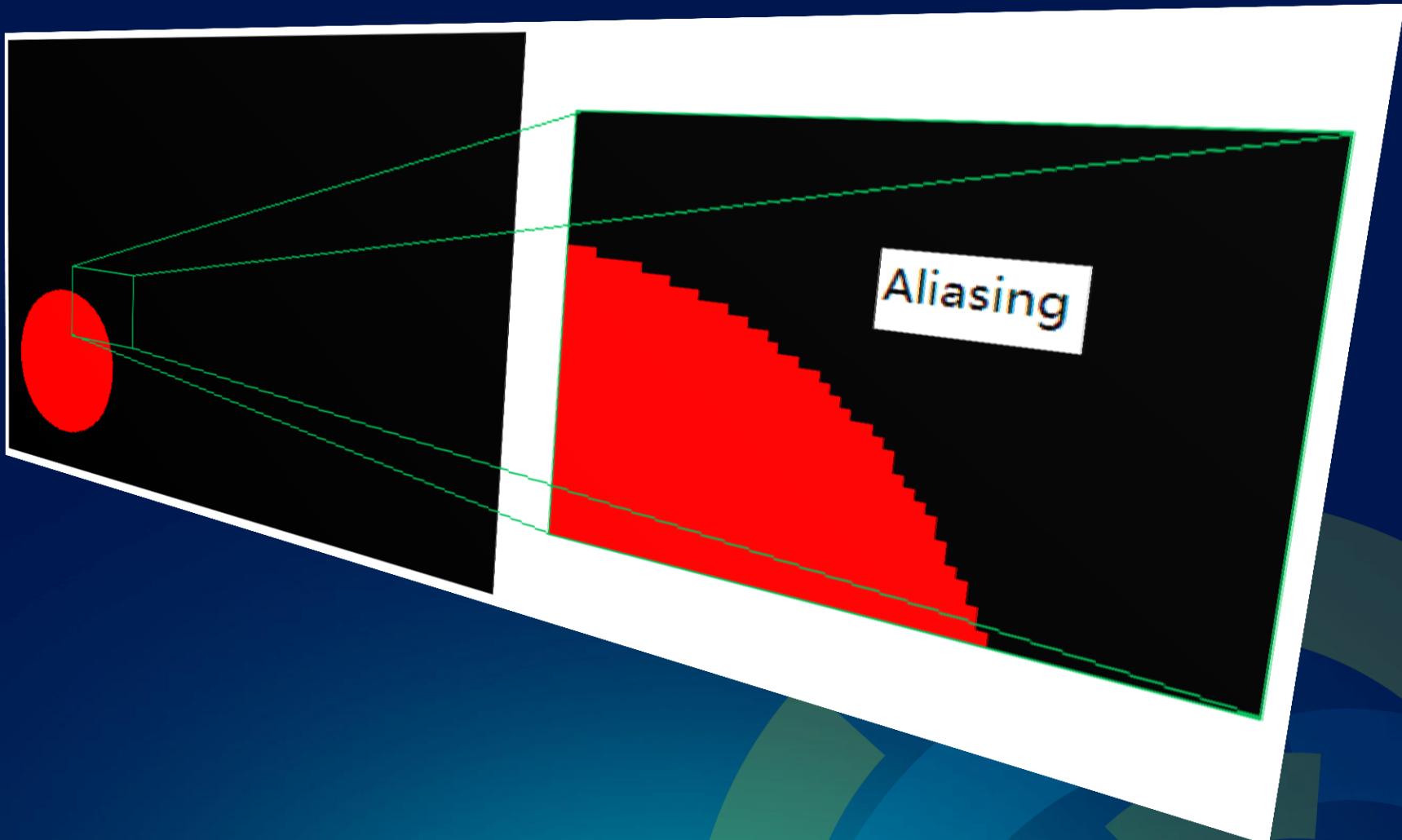
Pixel Manipulation

```
var setPixel = function (imageData, width, height, x, y, red,
                        green, blue) {
    var i = (y*width+x)*4;
    imageData[i] = red;
    imageData[i+1] = green;
    imageData[i+2] = blue;
    // NOT use Alpha: imageData[i+3]
}
// Since we use shape id as its pixel color, so:
// Color of Pixel of Hit on == hit shape id
var getPixelColor = function (x, y) {
    var pix = backendContext.getImageData(x, y, 1, 1).data;
    var r = pix[0]>>0;
    var g = pix[1]<<8;
    var b = pix[2]<<16;
    var a = pix[3]<<24;
    return (r|g|b|a);
};
```

How Anti-Alias disabled? Anti-Aliasing



Aliasing



Bresenham

- Bresenham for direct pixel manipulation
- Bresenham is fast but complex
- Bresenham aided elimination of anti-aliasing
(Bresenham辅助下禁止反走样)
- Built-in anti-aliasing disabling NOT work:
`anyContext.mozImageSmoothingEnabled = false;`
- We call for built-in implementation for anti-aliasing disabled.

JavaScript Programming - Rules

Fault:

var is optional

```
if (a>b)
    return;
if (c==d)
{
    return;
}
```

```
for (var i=0; i<10; i++) {
```

Right:

var is mandatory

```
if (a>b) {
    return;
}
if (c==d) {
    return;
}
```

```
var i;
for (i=0; i<10; i++) {
```

JavaScript Programming - Rules

`==`

`!=`

`if (a==b) ...`
`if (a!=b) ...`

`true`

`-----`
`0 == "`
`1 == true`
`0 == '0'`
`false == '0'`
`" \t\r\n " == 0`

`====`

`!==`

`if (a====b) ...`
`if (a!==b) ...`

`flase`

`-----`
`2 == true`
`false == 'false'`

JavaScript Check & Building Tools

- ✓ JRE
- ✓ JSLint
- ✓ YUICompressor
- ✓ NodeJS (curl, npm)
- ✓ Makefile
- ✓ Linux / Cygwin on Windows
- ✓ Git

JS Sample Project Downloads:

<https://github.com/Craga89/qTip2>

The screenshot shows the GitHub repository page for 'Craga89 / qTip2'. The URL 'https://github.com/Craga89/qTip2' is visible in the address bar. The repository is labeled as 'PUBLIC'. The 'Code' tab is selected. Below it, there is a brief description: 'qTip2 - Pretty powerful tooltips — Read more' and a link 'http://craigsworths.com/projects/qtip2'. There are download options: 'Clone in Windows', 'ZIP', 'HTTP', 'Git Read-Only', and 'https://github.com/Craga89/qTip2'. A dropdown menu shows the current branch is 'master'. Below the dropdown are 'Files' and 'Commits' tabs. A button says 'Download this repository as a zip file'. The GitHub logo is at the top left, and the page has a light gray background.

Project Structure

Craga89-qTip2-3785d11 ➤

 build

dist

src

.gitignore

 [GPL-LICENSE.txt](#)

Makefile

 MIT-LICENSE.txt

package.json

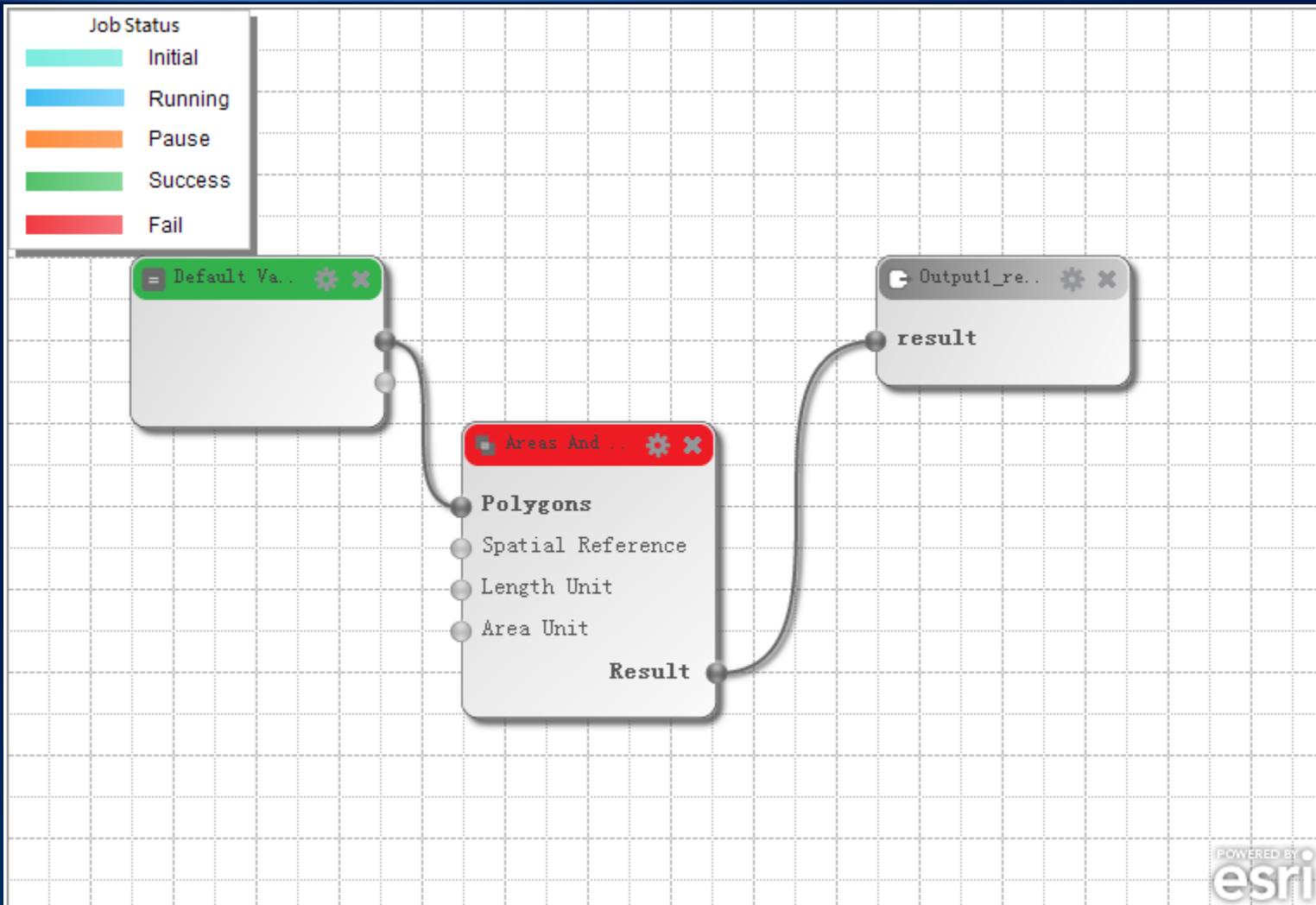
 README.md

 version.txt

Makefile

```
7 PLUGINS = $(shell ls -p ${SRC_DIR} | grep -v '^.$')
8 PLUGINS_JS = $(if ${PLUGINS},$(shell find ${SRC_DIR} -name ${PLUGINS} -print))
9 PLUGINS_CSS = $(if ${PLUGINS},$(shell find ${SRC_DIR} -name ${PLUGINS} -print))
10
11 EXTRA_CSS = ${SRC_DIR}/styles.css \
12                 ${SRC_DIR}/extra.css
13
14 JS_MODULES = ${SRC_DIR}/header.txt \
15                 ${SRC_DIR}/intro.js \
16                 ${SRC_DIR}/core.js \
17                 ${PLUGINS_JS} \
18                 ${SRC_DIR}/outro.js
19
20 CSS_MODULES = ${SRC_DIR}/header.txt \
21                 ${SRC_DIR}/core.css \
22                 ${PLUGINS_CSS} \
23                 ${EXTRA_CSS}
```

Demo: A Workflow Editor



Q&A

Thanks!

by

Liang_Zhang@esri.com

May. 22, 2012



esri China
BEIJING

分 享 地 理 价 值

