DEVELOPER CONFERENCE
MAY 7–9, 2012

HTML5 Canvas 2D Performance Tuning

Yang Gu, Yunchao He
Intel Open Source Technology Center

Agenda

- Canvas 2D 101
- Platform and Benchmark
- Performance Tuning
  - Platform-specific Optimization
  - Profile-Guided Hybrid Rendering
  - Immediate vs. Retained Mode
- Tips for Web App Developer

Canvas 2D 101 1/4

- An HTML5 element for scriptable 2D shapes and images
- History
  - Introduced in WebKit by Apple in 2004
  - Adopted in Gecko browser in 2005 and Opera in 2006
  - Standardized by WHATWG
  - Worked with W3C HTML Working Group

Canvas 2D 101 2/4

- Interface (CanvasRenderingContext2D)

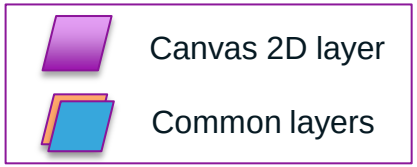
  State: save, restore
  Compositing: alpha, composite operation
  Color and style: stroke, fill, gradient, pattern
  Shadow: shadowOffset, shadowBlur, shadowColor
  Rect: clearRect, fillRect, strokeRect
  Path: beginPath, closePath, moveTo, lineTo, arcTo
  Text: font, align, baseline, fillText, strokeText, measureText
  Transformation: scale, rotate, translate, transform
  Line style: width, cap, join, miterLimit
  Image: drawImage, createImageData, getImageData

Canvas 2D 101 3/4

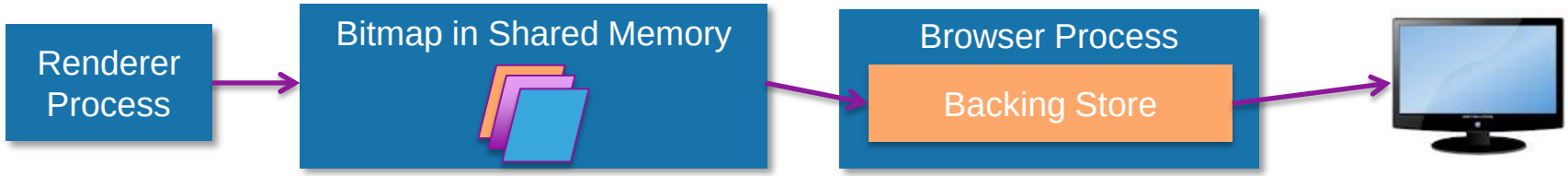
- Example

```
<canvas id="canvas" width="150" height="150"></canvas>
<script type="application/javascript">
    var canvas = document.getElementById("canvas");
    var ctx = canvas.getContext("2d");
    ctx.arc(20, 50, 10, Math.PI, Math.PI * 2, false);
    ctx.moveTo(70, 50);
    ctx.arc(80, 50, 10, Math.PI, Math.PI * 2, false);
    ctx.moveTo(30, 60);
    ctx.arc(50, 60, 20, Math.PI, Math.PI * 2, true);
    ctx.lineWidth = 5;
    ctx.strokeStyle = "red";
    ctx.stroke();
 </script>
```

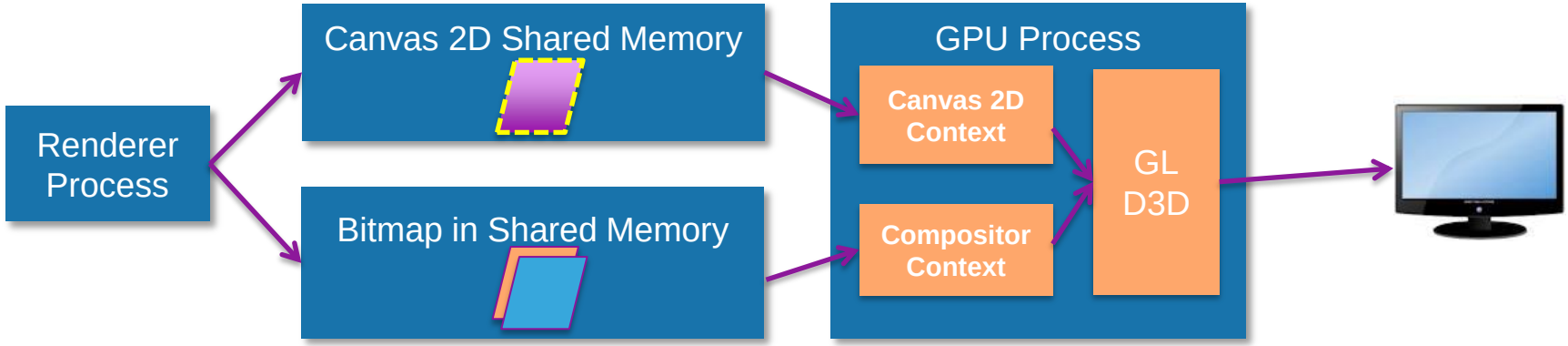
Canvas 2D 101 4/4



- Software rendering



- Hardware rendering



Platform

- Hardware
  - Desktop with cutting-edge CPU, GFX, memory, etc.
  - Netbook with relatively poor equipment
- Software
  - Ubuntu, Tizen
  - Chromium browser, with Skia as its 2D graphic library

Benchmark

- Selection process
  - Cover all feature aspects
  - Find well-known benchmarks first
  - Develop our own

Benchmark	Feature
Canvas Benchmark from MindCat	style, rect, path, text, transformation
Speed Reading from Microsoft	compositing, image
FishIE Tank from Microsoft	transformation, image
GUIMark 2 from Craftymind	path, line style
GUIMark 3 from Craftymind	rect, image
Internal benchmarks	compositing, shadow, …

Platform-specific Optimization 1/2

- Idea
  - Rewrite bottleneck functions with platform-specific instructions
- Example - benchmark “album”
  - Use SSE instructions to replace platform-neutral hot functions, such as SkARGB32_A8_BlitMask_SSE2, ClampX_ClampY_filter_affine_SSE2, S32A_Opaque_BlitRow32_SSE2, etc.
  - Provides around a 20% performance gain in software rendering
- More optimizations at third_party/skia/src/opts/

Platform-specific Optimization 2/2

- Best Practice
  - Run each benchmark with bad performance in browser
  - Trace renderer process to find bottleneck (perf is our friend here)
  - If any, bottlenecks typically exist in graphic library function
  - Optimize bottleneck with platform-specific instructions

Profile-Guided Hybrid Rendering 1/2

- Idea
  - Instead of pure software rendering or hardware rendering, use hybrid rendering
  - Switch rendering mode based on profile
- Impact factor - interface
  - HW wins on transformation (scale, rotate)
  - SW wins on most other parts

Cases in featureBatchTest	SW	HW	Comparision
hline	159	52.0	-67.3%
vline	130	51.6	-60.3%
line	57.3	48.4	-15.5%
rect	93.6	41.9	-55.2%
fill_rect	37.3	19.3	-48.3%
lines	83.9	88.9	5.95%
arc	42.7	30.2	-29.3%
fill_arc	24.0	20.1	-16.3%
bezier	29.4	29.7	-1.02%
fill_bezier	12.0	5.27	-56.1%
quad	39.9	29.5	-26.1
curves	64.7	57.8	-10.7%
fill_curves	31.0	20.3	-34.5%
stroke_star	19.4	13.8	-28.9%
fill_star	30.0	15.1	-49.7%
transform	0.672	0.833	24.0%
image	15.7	6.07	-61.3%
image_scale	6.30	8.73	38.6%
image_rotate	0.709	1.35	90.4%
linear_gradient	11.3	7.42	-34.3%
radial_gradient	4.08	6.95	70.3%
text	16.4	11.1	-32.3%
clip	8.45	5.81	-31.2%
Total result	5.82	3.95	-32.1%

Profile-Guided Hybrid Rendering 2/2

- Impact factor – Canvas size

Canvas Size	SW	Rate	HW	Rate
480 * 480	4.3	100%	1.43	100%
960 * 960	1.77	41.2%	0.939	65.7%
1440 * 1440	0.948	22.0%	0.615	43.0%

- Impact factor – Switch overhead
  - SW->HW: Upload the bitmap as texture to GPU Process
  - HW->SW: Download texture to Renderer Process to draw

Immediate vs. Retained Mode 1/3

- Idea
  - Current canvas operations are all in immediate mode, we may add support for retained mode
- Mode comparison

Mode	Description	Pro	Con
Immediate	Client calls directly cause rendering of graphic objects	Full control, debug-ability	Expensive for CPU and GPU
Retained	Instead of direct rendering, update the internal state of graphic library	Better performance	No full control, not easy to debug

Immediate vs. Retained Mode 2/3

- Example – Combine operations

```
for (var i = 0; i < 100; i++) {
  var p1 = p[i];
  var p2 = p[i+1];
  context.beginPath();
  context.moveTo(p1.x, p1.y);
  context.lineTo(p2.x, p2.y);
  context.stroke();
}
```

1611 iterations / second

```
context.beginPath();
for (var i = 0; i < 100; i++) {
  var p1 = p[i];
  var p2 = p[i+1];
  context.moveTo(p1.x, p1.y);
  context.lineTo(p2.x, p2.y);
}
context.stroke();
```

20942 iterations / second

Immediate vs. Retained Mode 3/3

- Example – Avoid frequent style change

```
for (var i = 0; i < 100; i++) {
  context.fillStyle = (i % 2 ? “red” : “blue”);
  context.fillRect(i, 0, 1, 480);
}
```

1739 iterations / second

```
context.fillStyle = “red”;
for (var i = 0; i < 50; i++) {
  context.fillRect(i * 2, 0, 1, 480);
}
context.fillStyle = “blue”;
for (var i = 0; i < 50; i++) {
  context.fillRect(i * 2 + 1, 0, 1, 480);
}
```

2214 iterations / second

Tips for Web App Developer 1/3

- Multiple-layered canvas
  - Avoid redrawing big images frequently
- Example

```
<canvas id="background" style="z-index: 0"></canvas>
<canvas id="foreground"  style="z-index: 1"></canvas>
```

Tips for Web App Developer 2/3

- requestAnimationFrame – JS based animation
  - Not invoked if:
    - tab is not visible
    - FPS > 60
    - Faster than page rendering
- Example

```
function drawFrame() {
        ...
        requestAnimationFrame(drawFrame);
}
drawFrame();
```

Tips for Web App Developer 3/3

- Proper way to clear the canvas
  - canvas.width = canvas.width;
    - Not only clears the canvas, but also all states: transformation matrix, clipping region, fillStyle, globalAlpha, etc.
    - Slower
  - ctx.clearRect(0, 0, canvas.width, canvas.height);
    - Only clears the canvas
    - Typically faster

Questions?

TIZEN™ DEVELOPER CONFERENCE
MAY 7–9, 2012