

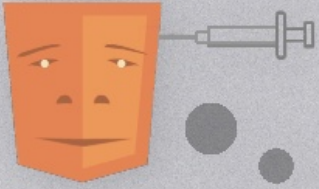


Optimiser le <canvas>

par #ponce (gamesfrommars.fr)

via wullon ([adinpsz](http://adinpsz.com), demoscene.fr)

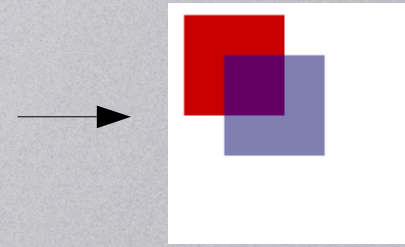
• HTML CANVAS OPTIMIZATIONS •



L'objet HTMLCanvas

(source : developer.mozilla.org)

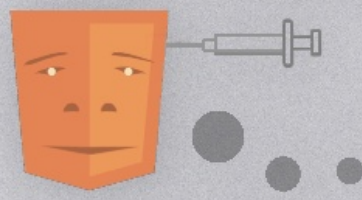
```
1 <html>
2   <head>
3     <script type="application/javascript">
4       function draw() {
5         var canvas = document.getElementById("canvas");
6         var ctx = canvas.getContext("2d");
7
8         ctx.fillStyle = "rgb(200,0,0)";
9         ctx.fillRect (10, 10, 55, 50);
10
11        ctx.fillStyle = "rgba(0, 0, 200, 0.5)";
12        ctx.fillRect (30, 30, 55, 50);
13      }
14    </script>
15  </head>
16  <body onload="draw()">
17    <canvas id="canvas" width="300" height="300"></canvas>
18  </body>
19 </html>
```



browser

Javascript

HTML CANVAS OPTIMIZATIONS



API 2D

- Privilégie la simplicité
- Conçue pour le rendu soft

HTML5 Canvas Cheat Sheet v1.1 <http://blog.nihilogic.dk/>

| Canvas element | | |
|---|---|---------|
| Attributes | | |
| Name | Type | Default |
| width | unsigned long | 300 |
| height | unsigned long | 150 |
| Methods | | |
| Return | Name | |
| string | toDataURL([Optional] string type, [Variadic] any args) | |
| Object | getContext(string contextId) | |
| 2D Context | | |
| Attributes | | |
| Name | Type | |
| canvas | HTMLCanvasObject [readonly] | |
| Methods | | |
| Return | Name | |
| void | save() | |
| void | restore() | |
| Transformation | | |
| Methods | | |
| Return | Name | |
| void | scale(float x, float y) | |
| void | rotate(float angle) | |
| void | translate(float x, float y) | |
| void | transform(float m11, float m12, float m21, float m22, float dx, float dy) | |
| void | setTransform(float m11, float m12, float m21, float m22, float dx, float dy) | |
| Image drawing | | |
| Methods | | |
| Return | Name | |
| void | drawImage(Object image, float dx, float dy, [Optional] float dw, [float dh]) | |
| Argument 'image' can be of type HTMLImageElement, HTMLCanvasElement or HTMLVideoElement | | |
| void | drawImage(Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh) | |

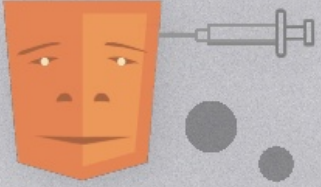
| Compositing | | |
|---------------------------------------|------------------|-------------|
| Attributes | | |
| Name | Type | Default |
| globalAlpha | float | 1.0 |
| globalCompositeOperation | string | source-over |
| Supports any of the following values: | | |
| | source-over | |
| | source-in | |
| | source-out | |
| | source-atop | |
| | destination-over | |
| | destination-in | |
| | destination-out | |
| | lighter | |
| | copy | |
| | xor | |
| Line styles | | |
| Attributes | | |
| Name | Type | Default |
| lineWidth | float | 1.0 |
| lineCap | string | butt |
| Supports any of the following values: | | |
| | butt | |
| | round | |
| | square | |
| | miter | |
| Supports any of the following values: | | |
| | round | |
| | bevel | |
| | miter | |
| | miterLimit | 10 |

| Colors, styles and shadows | | |
|--|--|-------------------|
| Attributes | | |
| Name | Type | Default |
| strokeStyle | string | black |
| fillStyle | any | black |
| shadowOffsetX | float | 0.0 |
| shadowOffsetY | float | 0.0 |
| shadowBlur | float | 0.0 |
| shadowColor | string | transparent black |
| Methods | | |
| Return | Name | |
| CanvasGradient | createLinearGradient(float x0, float y0, float x1, float y1) | |
| CanvasGradient | createRadialGradient(float x0, float y0, float r0, float x1, float y1, float r1) | |
| CanvasPattern | createPattern(Object image, string repetition) | |
| Argument 'image' can be of type HTMLImageElement, HTMLCanvasElement or HTMLVideoElement | | |
| 'repetition' supports any of the following values: [repeat (default), repeat-x, repeat-y, no-repeat] | | |
| CanvasGradient interface | | |
| void | addColorStop(float offset, string color) | |
| CanvasPattern interface | | |
| No attributes or methods. | | |
| Paths | | |
| Methods | | |
| Return | Name | |
| void | beginPath() | |
| void | closePath() | |
| void | fill() | |
| void | stroke() | |
| void | clip() | |
| void | moveTo(float x, float y) | |
| void | lineTo(float x, float y) | |
| void | quadraticCurveTo(float cp1x, float cp1y, float cp2x, float cp2y, float x, float y) | |
| void | bezierCurveTo(float cp1x, float cp1y, float cp2x, float cp2y, float x, float y) | |
| void | arc(float x, float y, float radius) | |
| void | arc(float x, float y, float radius, float startAngle, float endAngle, boolean anticlockwise) | |
| void | rect(float x, float y, float w, float h) | |
| boolean | isPointInPath(float x, float y) | |


| Text | | |
|---|---|-----------------|
| Attributes | | |
| Name | Type | Default |
| font | string | 10px sans-serif |
| textAlign | string | start |
| Supports any of the following values: [start, end, left, right, center] | | |
| textBaseline | string | alphabetic |
| Supports any of the following values: [top, hanging, middle, alphabetic, ideographic, bottom] | | |
| Methods | | |
| Return | Name | |
| void | fillText(string text, float x, float y, [Optional] float maxWidth) | |
| void | strokeText(string text, float x, float y, [Optional] float maxWidth) | |
| TextMetrics | measureText(string text) | |
| TextMetrics interface | | |
| width | float | [readonly] |
| Rectangles | | |
| Methods | | |
| Return | Name | |
| void | clearRect(float x, float y, float w, float h) | |
| void | fillRect(float x, float y, float w, float h) | |
| void | strokeRect(float x, float y, float w, float h) | |
| Pixel manipulation | | |
| Methods | | |
| Return | Name | |
| ImageData | createImageData(float sw, float sh) | |
| ImageData | createImageData(ImageData) | |
| ImageData | getImageData(float sx, float sy, float sw, float sh) | |
| void | putImageData(ImageData, float dx, float dy, [Optional] float dirtyX, [float dirtyY, float dirtyWidth, float dirtyHeight]) | |
| ImageData interface | | |
| width | unsigned long | [readonly] |
| height | unsigned long | [readonly] |
| data | CanvasPixelArray | [readonly] |
| CanvasPixelArray interface | | |
| length | unsigned long | [readonly] |

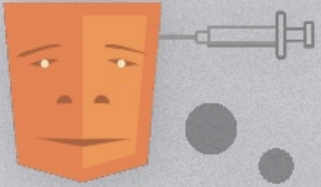
Source: <http://www.whatwg.org/specs/web-apps/current-work/> (2009-05-04)

Un canvas ≈ une texture ≈ un framebuffer



Pourquoi utiliser le <canvas> ?

- Support : 55% des browsers (source : caniuse.com)
- Soit 1,1 milliards d'internautes 
- Parce qu'on peut
- **Portabilité** Win/Mac/Linux/smartphones



Et pourquoi pas WebGL ?

Support WebGL :

| IE | Firefox | Safari | Chrome | Opera | iOS Safari | Opera Mini | Opera Mobile | Android Browser | |
|-----|---------|--------|--------|-------|------------|------------|--------------|-----------------|-----|
| 9.0 | 4.0 | 5.0 | 11.0 | 11.1 | 4.2-4.3 | 5.0-6.0 | 11.0 | 2.3 | 3.0 |

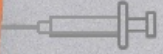
Support <canvas> :

| IE | Firefox | Safari | Chrome | Opera | iOS Safari | Opera Mini | Opera Mobile | Android Browser | |
|-----|---------|--------|--------|-------|------------|------------|--------------|-----------------|-----|
| 9.0 | 4.0 | 5.0 | 11.0 | 11.1 | 4.2-4.3 | 5.0-6.0 | 11.0 | 2.3 | 3.0 |

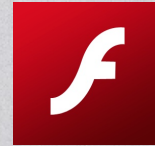
(source : caniuse.com)



- plus puissant
- support limité pour le moment

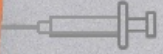


Et Flash ?



- + popularité > 90%
- + implémentation plus uniforme
- + meilleurs outils
- c'est Flash

Eviter le Canvas si on a le choix !



Cas d'utilisations ?

- Jeu vidéo portable
- Application graphique Web (tools ?)
- Compétitions :
 - JS1k
 - 10k apart
 - Mozilla Game On
 - Microsoft {Dev:unplugged}
 - HTML5contest.com
 - etc...



Le problème avec `<canvas>` :

C'est lent et mal spécifié.



Le problème avec `<canvas>` :

C'est lent et mal spécifié.

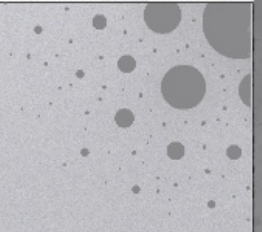
Javascript

Routines de dessins

double-buffering ?

filtrage ?

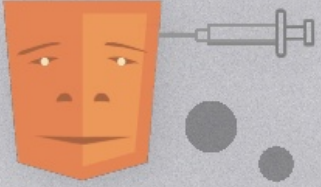
espace couleur ?



Javascript

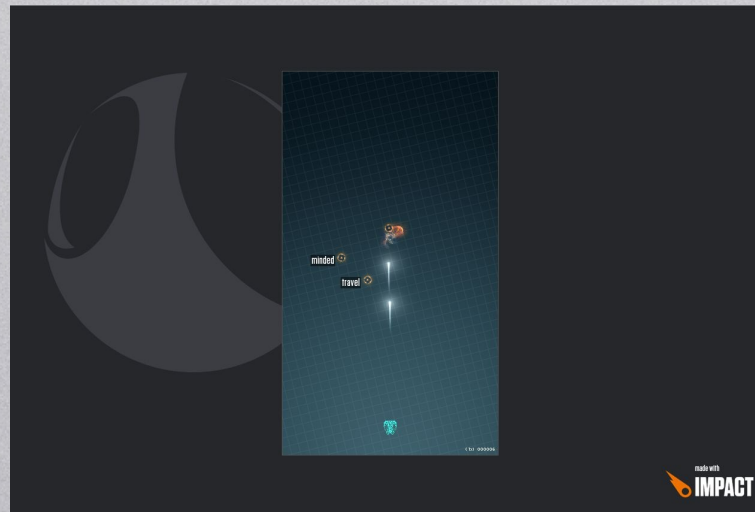
- Pauses GC si on ne fait pas gaffe (eg : ro.me)
- Constructions à éviter
- blog.gamesfrommars.fr pour quelques tips



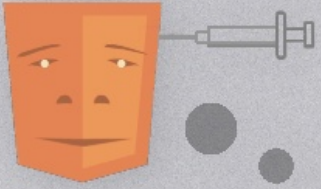


Routines de dessins

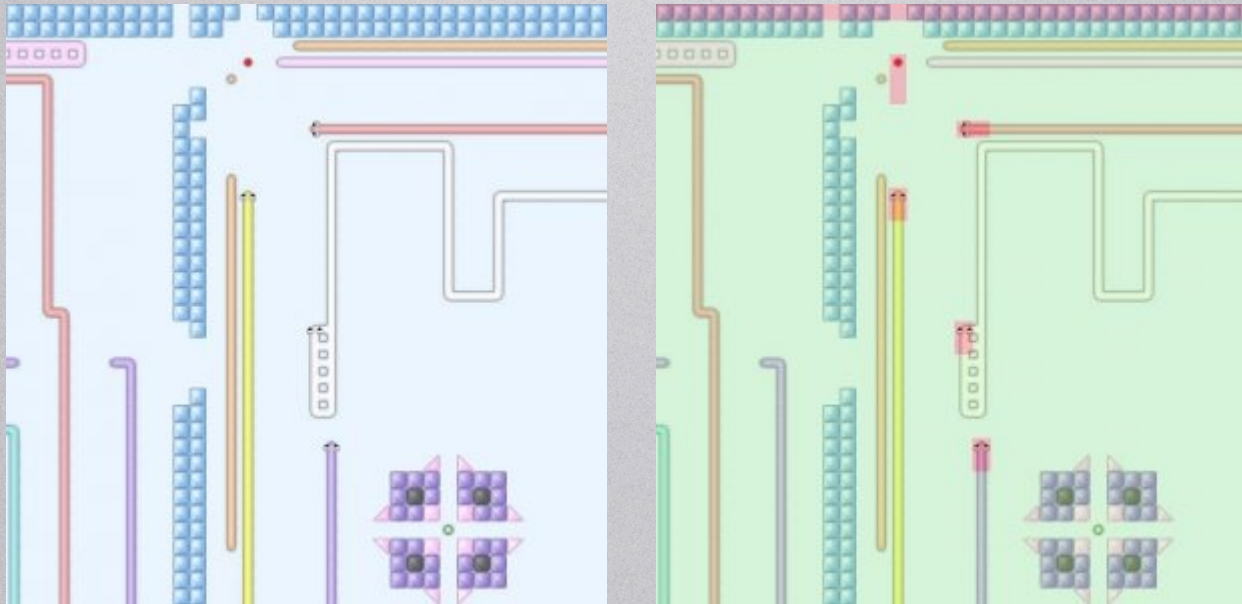
- Dominent les benchmarks
- Même optimisé au maximum
- Pas facile d'avoir une grosse surface de jeu



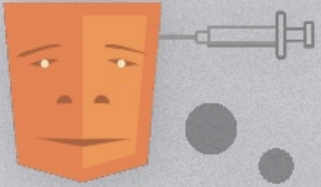
Z-Type = 360x640 pixels ॐ_ॐ



Optimiser un rendu tilé



- Réutiliser la frame précédente
- La décaler en sens inverse de la camera (enfin... vous voyez)
- Dessiner les différences qui restent à base de blit

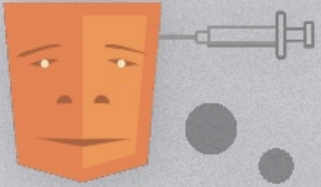


Composer plusieurs canvas



Canvas Rider = gros niveau pré-rendu + sprites

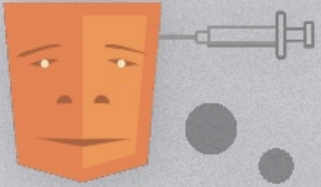
Browser compositing **plus rapide** que le blit du canvas !



Utiliser des primitives vectorielles



TankWorld = rendu 3D flat
Faire des batchs par couleur



Filtrage

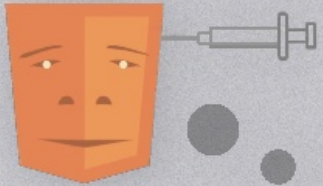
Les images sont filtrées au blit => précalculer la bonne taille



Biolab Disaster version hq2x

Choix du filtre nearest ou « autre » pour une image :

```
15   var ieValue = quality ? "bicubic" : "nearest-neighbor";
16   var ffValue = quality ? "optimizeQuality" : "optimizeSpeed";
17
18   if (img.style.msInterpolationMode !== undefined) {
19     img.style.msInterpolationMode = ieValue;
20   }
21   else if (img.style.getPropertyValue("image-rendering") != null)
22   {
23     img.style.setProperty ("image-rendering", ffValue, null);
24   }
```

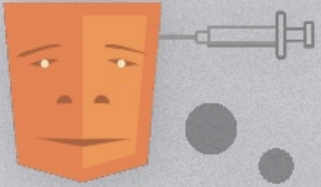
Filtrage

Le `<canvas>` est **également** filtré lors du compositing.

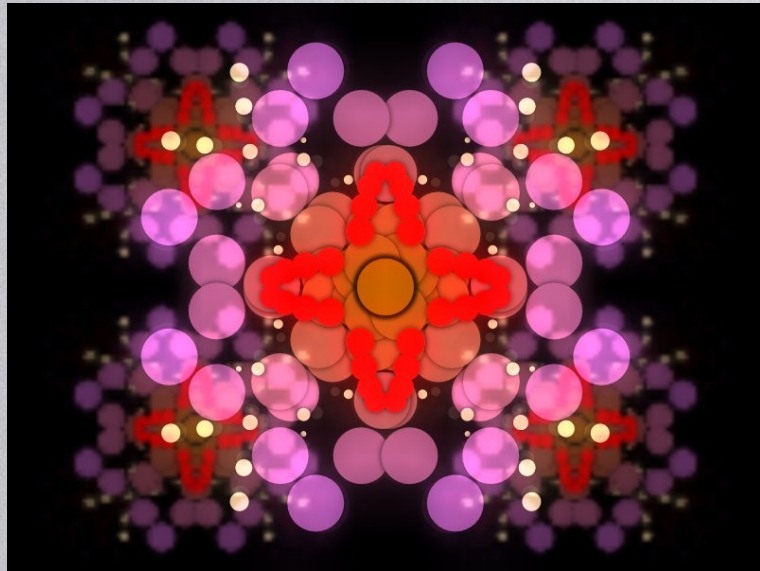
```
1 if (canvas.style.msInterpolationMode !== undefined) {  
2   canvas.style.msInterpolationMode = "nearest-neighbor";  
3 }  
4 else if (this._canvas.style.getPropertyValue("image-rendering") != null)  
5 {  
6   canvas.style.setProperty("image-rendering", "optimizeSpeed", null);  
7 }
```

Code qui accélère tout

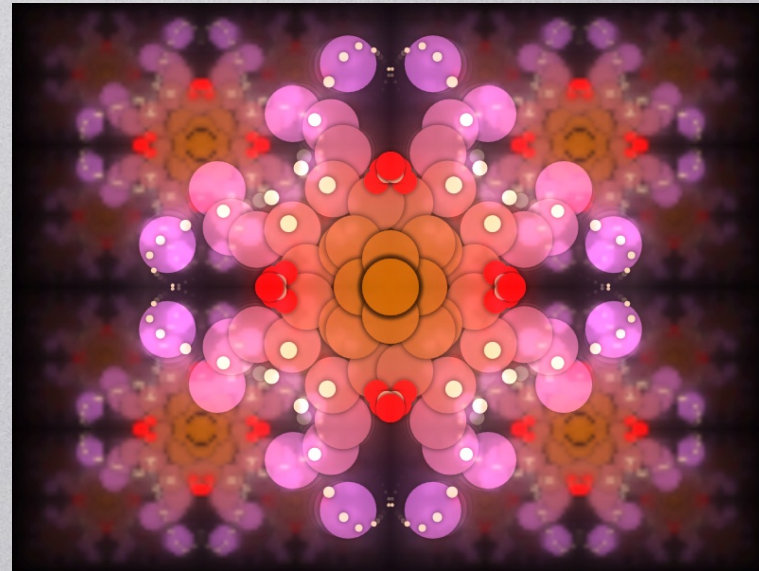
- Filtrage nearest améliore beaucoup les perfs
- Affecte le rendu de texte + certaines primitives
- Ni dans la norme, ni dans la doc !



Différences d'implémentations



Opera 11

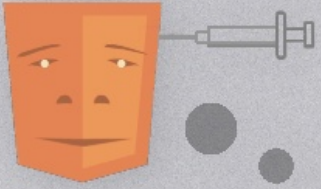


Chrome 11

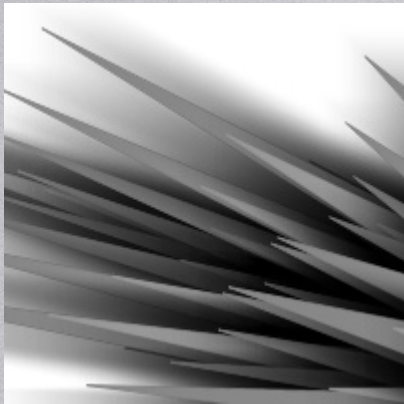
Le blending varie selon les browsers !

Attention si vous cascadez des blits.

(ici blur pyramidal)



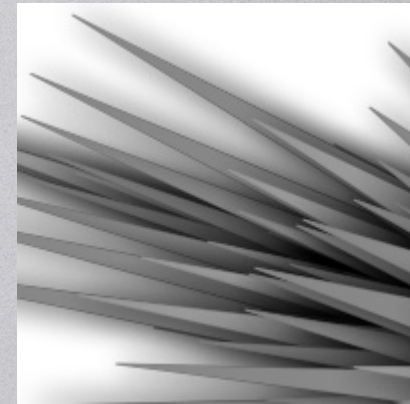
Différences d'implémentations



Chrome 11

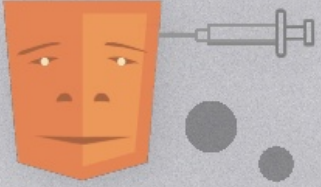


Firefox 4



Opera 11

`shadowBlur` pas implémenté pareil.



Outils

- **Firebug** : profiler
- **stats.js bookmark** : pour mesurer le framerate
- **jsperf.com** : benchmarks Javascripts
- **jsbeautify** : pour réverser le code des autres
- Pour l'obfuscation : **YUIcompressor** marche bien
- **excanvas.js** : librairie canvas-like pour IE7/IE8 (très lente)



Questions ?

contact@gamesfrommars.fr
bbs.demoscene.fr