

株式会社 ITS MORE

2020年4月設立

2020年11月8日 投稿者: SATOXITS

GShell 0.8.0 – SVG事始め

社長：いやー、1週間ぽっかり休みましたね。

基盤：大統領選がきっかけですっかりSNSにはまりました。

社長：でも、おかげでいろいろヒントを得ました。

開発：脳味噌がすっからかーんな感じになりましたwある種の爽快感。

社長：SVGをやりかけて止まりましたので、あれから再開しましょうか。

* * *

開発：それではまずスマイルマークより。丸だけでお試し。タグの書き方が2とおりあるようなので、両方で。

```
<div>
<svg id="SVG_01" width="100" height="100">
<circle cx="50" cy="50" r="40" stroke="white" stroke-width="0" fill="yellow" />
<circle cx="35" cy="35" r="05" stroke="white" stroke-width="0" fill="#000000ff" />
<circle cx="65" cy="35" r="05" stroke="white" stroke-width="0" fill="#000000ff" />
<circle cx="50" cy="60" r="20" stroke="black" stroke-width="2" fill="#00000000" />
</svg>

<svg id="SVG_02" viewBox="0 0 100 100">
<circle cx="50" cy="50" r="40" stroke="white" stroke-width="0" fill="yellow" />
<circle cx="35" cy="35" r="05" stroke="white" stroke-width="0" fill="#000000ff" />
<circle cx="65" cy="35" r="05" stroke="white" stroke-width="0" fill="#000000ff" />
<circle cx="50" cy="60" r="20" stroke="black" stroke-width="2" fill="#00000000" />
</svg>
</div>
```



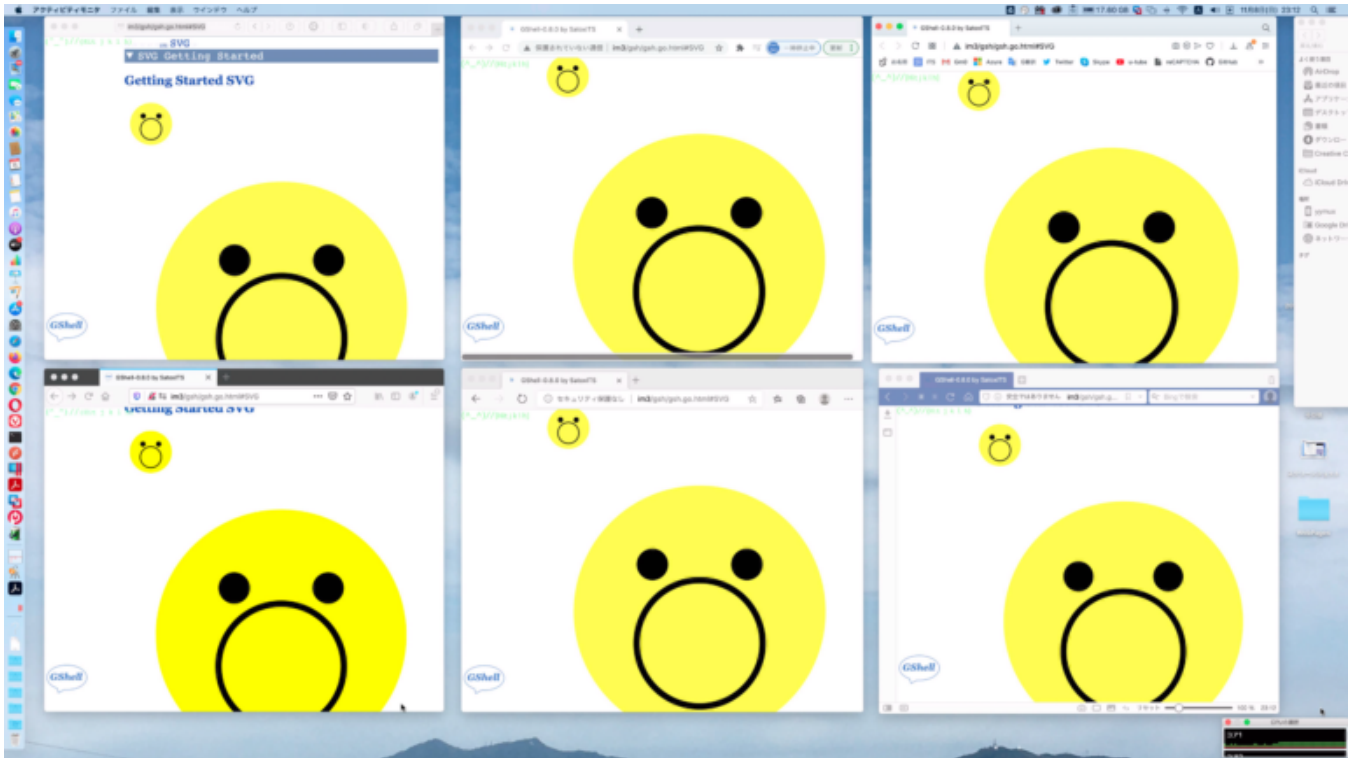
基盤：おおっと、viewBoxのほうは顔がでかい！

開発：周囲との比率指定みたいです。とりあえず CSSで枠を決めます。



社長：どっちの書き方にすべきなのでしょうね？

開発：どのブラウザもどっちでも良いみたいですが。MDNに書いてある方法のほうがきっと将来的には良いのかな… 後で調べましょう。



基盤：まずは口ですね。口を半円か円弧にしましょうよ。

開発：そうですね。えーと円弧の角度…

* * *

開発：どうもよくわかりませんが、canvas の arc に相当するものはないですね。path で書くみたいなんですけど、始点と終点、あと曲率？を指定するみたいなの。

```
<svg id="SVG_03" class="SVG100" viewBox="0 0 100 100">
<circle cx="50" cy="50" r="40" stroke="white" stroke-width="0" fill="yellow" />
<circle cx="35" cy="35" r="05" stroke="white" stroke-width="0" fill="#000000ff" />
<circle cx="65" cy="35" r="05" stroke="white" stroke-width="0" fill="#000000ff" />
<path fill="orange" d="M 25 50 A 10 10 0 0 0 75 50 Z"/>
</svg>
```

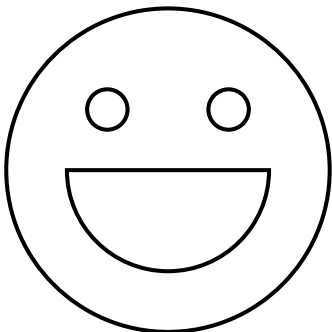
社長：もろに昔のプロッターのコマンドみたいな指示方法ですね。



社長、基盤：おおーっ。

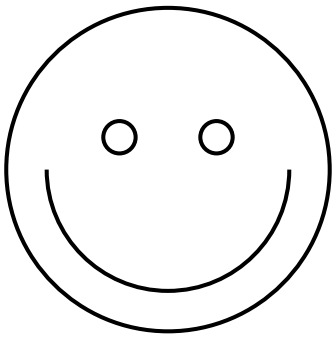
社長：canvas で描いたスマイルマークと同じものを書くとうなるでしょう？

開発：まず、輪郭だけにするとこうなりますね。



開発：で、顔のパーツの位置を適当に調整するとこういう感じ。

```
<svg id="SVG_04" class="SVG100" viewBox="0 0 100 100">
<circle cx="50" cy="50" r="40" stroke="black" stroke-width="1" fill="#ffffff00" />
<circle cx="38" cy="42" r="04" stroke="black" stroke-width="1" fill="#00000000" />
<circle cx="62" cy="42" r="04" stroke="black" stroke-width="1" fill="#00000000" />
<path stroke="black" d="M 20 50 A 10 10 0 0 0 80 50" fill="#ffffff00"/>
</svg>
```

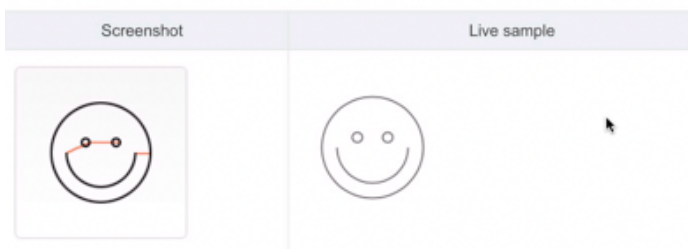


社長、基盤：おおーっ、スマイルマークだ。

開発：ちなみにMDNのオリジナルのスマイルマークがこれ。

```
1 function draw() {
2   var canvas = document.getElementById('canvas');
3   if (canvas.getContext) {
4     var ctx = canvas.getContext('2d');
5
6     ctx.beginPath();
7     ctx.arc(75, 75, 50, 0, Math.PI * 2, true); // Outer circle
8     ctx.moveTo(110, 75);
9     ctx.arc(75, 75, 35, 0, Math.PI, false); // Mouth (clockwise)
10    ctx.moveTo(65, 65);
11    ctx.arc(60, 65, 5, 0, Math.PI * 2, true); // Left eye
12    ctx.moveTo(95, 65);
13    ctx.arc(90, 65, 5, 0, Math.PI * 2, true); // Right eye
14    ctx.stroke();
15  }
16 }
```

he result looks like this:



開発：まあ、計算して出せば、ピクセル単位でまったく同じものができると思いますが。

社長：SVG に canvas 互換関数で書くというような標準関数はないんでしょうかね？

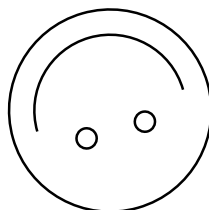
開発：標準では無いように見えますが・・・

基盤：回転してみたいですね。

開発：これは普通に、CSSで回せば良いのかなと思います。



Canvas ver.



SVG ver.

開発：うーん、Safari / iMac で 30ms 間隔ですが、回転させるのにまずタイマーイベント処理に2%、canvas だとCPUを 2%、SVG だと 7% くらい食ってるように見えますね。

基盤：タイマーイベント処理の部分はCSSアニメ機能を使えば軽くなりそうです。

社長：SVGが重いのは、構成してるエレメントの位置とかを再計算してるからでしょうかね。

開発：SVGだと拡大縮小しても最適な表示が描画されるわけですけど、それは canvas だってリサイズに応じてパラメータを変えて再描画すればいいわけで、それ自体はSVGのメリットとは言い難い感じはします。

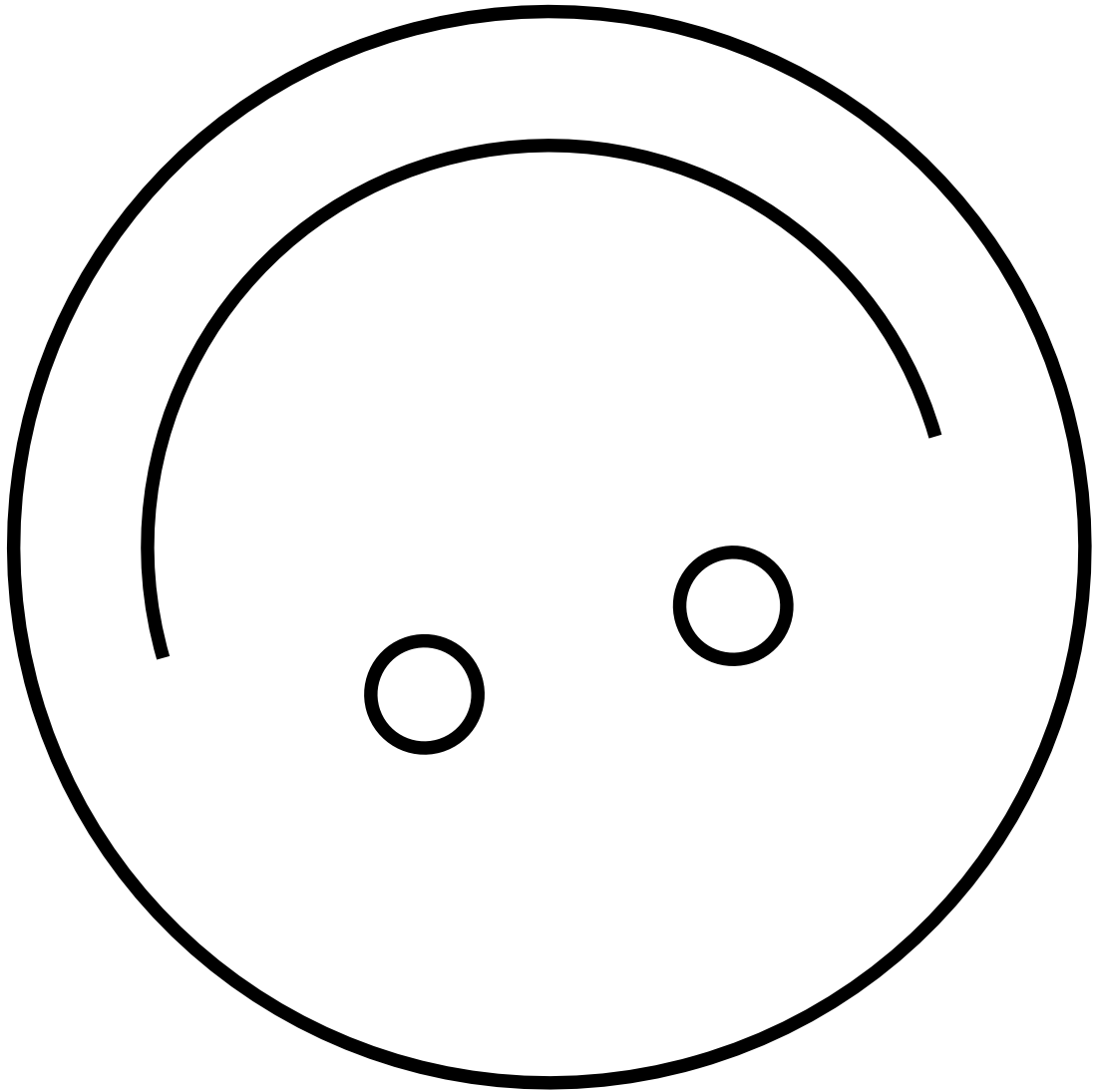
社長：やはり、パーツ単位で移動できたり、クリックしてイベントを発生させたりするようなアプリに向いているということでしょうね。

Canvas ver.

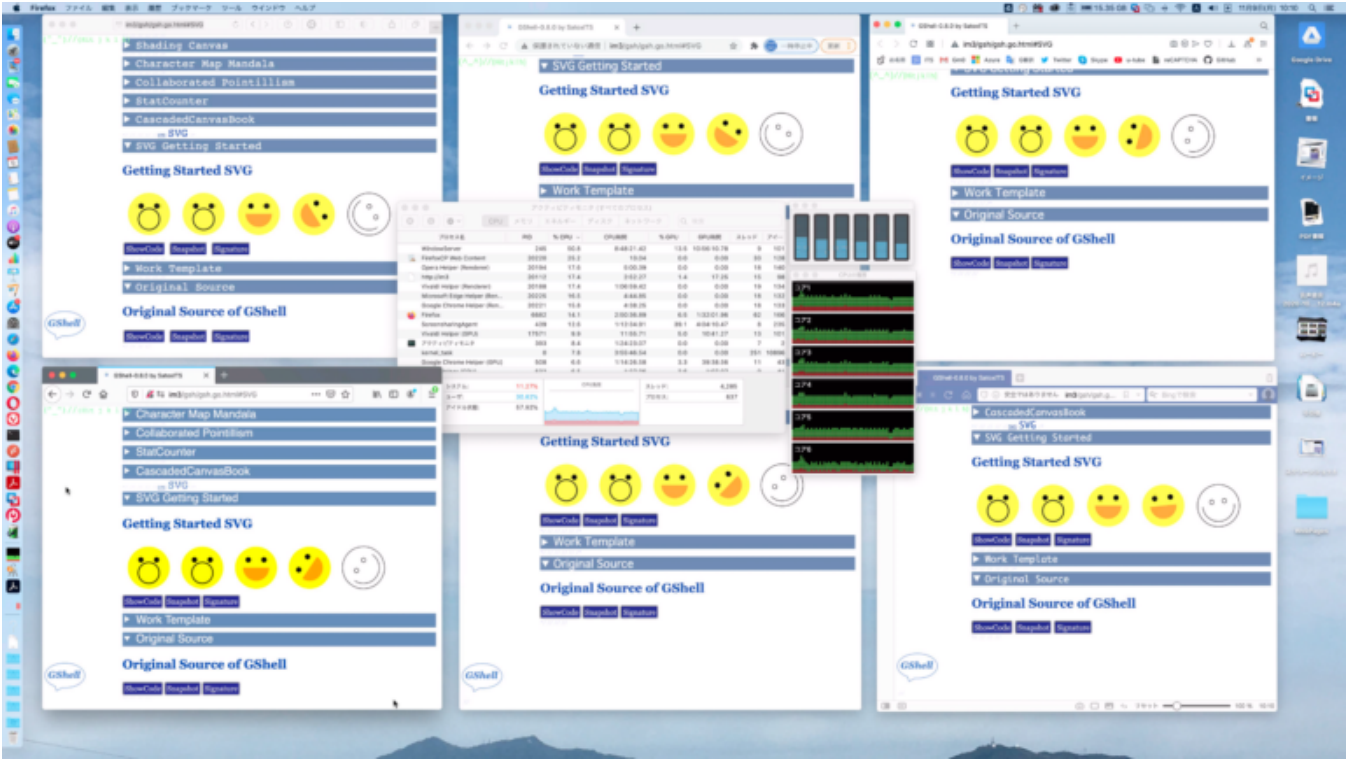


ver.

SVG



基盤：これ、実行するとなぜかFirefoxだけ重いように見えます。



社長：色々個性があって面白いですね。

-- 2020-11-08 SatoxITS

gsh-0.8.0.go [ダウンロード](#)

GShell-o.8.0 by SatoxITS

UA:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0 Safari/605.1.15

2020/10/09 10:20:26

2020/10/09 10:21:12

Elapsed: 46.222 s

Skew: 0 ms

GShell version 0.8.0 // 2020-11-08 // SatoxITS

Shell \Rightarrow **GShell** \Rightarrow **GSI**

- * / / / / *
- ▶ Topbar
- * / / / / / / / *
- ▶ Indexer
- * / / / / *

GShell // a General purpose Shell built on the top of Golang

It is a shell for myself, by myself, of myself. --SatoxITS(^-^)
prev.

Edit Save Load Vers 0 Fork Stop Unfold
 Digest Source

- [▶ Statement](#)
- [▶ Features](#)
- [▶ Index](#)
- [▶ Go Source](#)
- [▶ Considerations](#)
- [▶ References](#)
- [▶ Raw Source](#)



(^_^)//{Hit j k l h}

- [▶ PackmonGo](#)
- [▶ SightGlass](#)
- [▶ GJ Console](#)
- [▶ Form Auto. Filling](#)
- [▶ BlinderText](#)
- [▶ Golang / JavaScript Link](#)
- [▶ GJ Link](#)
- [▶ Live HTML Snapshot](#)



