

# WebGL によるデータ可視化入門\*

## アニメーションと演習

陰山 聡

神戸大学 システム情報学研究科 計算科学専攻

2014.07.01

アニメーション

演習

# アニメーション

## 二つの方法

1. JavaScript の `setInterval()` を使う
2. `requestAnimationFrame()` を使う

## setInterval

```
setInterval(codeToCall, timeoutInMilliseconds)
```

使い方：

```
function draw() {  
  . . .  
}
```

```
function start() {  
  . . .  
  setInterval(draw, 16.7);  
}
```

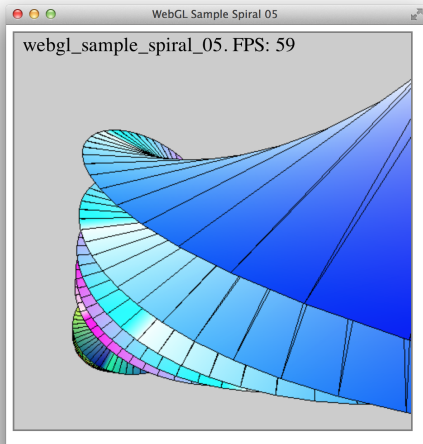
16.7 ミリ秒<sup>†</sup>毎に draw() を呼ぶ。

---

<sup>†</sup>1/60 ~ 0.0167

## setInterval を使ったサンプルコード

webgl\_sample\_spiral\_05.html

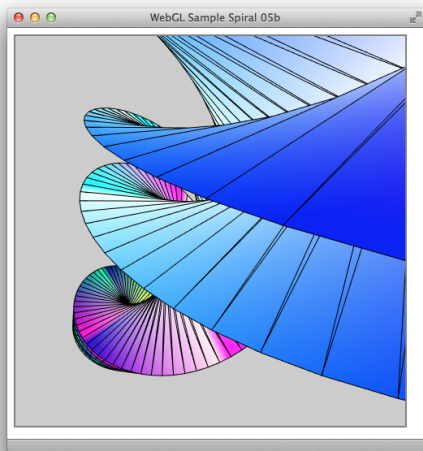


## requestAnimationFrame

- シーンの更新タイミングをブラウザに任せる。
- こちらのほうがよい。

# requestAnimationFrameを使ったサンプルコード

webgl\_sample\_spiral\_05b2.html





## requestAnimationFrame の使い方

引数にブラウザから呼び出すコールバック関数を渡す。

```
requestAnimationFrame(draw);
```

指定したコールバック draw 関数が呼び出されるときには、引数として現在時刻 (currentTime) が自動的に渡される。

つまり、draw() 関数は currentTime を引数として受け取るように定義する。

ただし、startup 関数から最初に draw() を呼び出すときには引数なしでコールする。(JavaScript では文法エラーにならない。)

## requestAnimationFrame の使い方

```
function draw(currentTime) {  
  requestAnimationFrame(draw);  
  . . .  
}
```

```
function startup() {  
  . . .  
  draw();  
}
```

```
function draw(currentTime) {  
  // 1. 現在のフレーム描画を開始する前に次のフレームを  
  //    描画する新たな呼び出しを要求  
  
  requestAnimationFrame(draw);  
  
  // 2. シーンの中で動くオブジェクトの位置を更新  
  ...  
  // 3. シーンを描画
```

# 演習

# 演習

- `webgl_sample_spiral_05b2.html` を自由に変更し、アニメーションで遊んでみよう。