

CL Server勉強会

kayac 川崎

2017/02/27

アジェンダ

- Intro
- 本の紹介
- 使用上の注意
- まとめ

Intro

CanvasとCreateJS

Canvas API

<http://www.html5.jp/canvas/ref.html>

- canvas要素に全て吸収されるのでDOMイベントの設定が難しい
(座標とかで頑張れば)
- オブジェクトの操作も大変 (状態管理など)

CreateJS

<http://www.createjs.com/docs>

- Canvasを便利に使うライブラリ群
- 前のスライドの問題点は大体解決

サポートブラウザ

<http://caniuse.com/#search=canvas>

CreateJS libraries inheritance approach changed in EaselJS 0.8, TweenJS 0.6, PreloadJS 0.6, and SoundJS 0.6 is not compatible with IE8 and below. For library support for earlier versions of IE, use an older version of the libraries, available as tags/releases in GitHub, and on the CreateJS CDN. EaselJS is only compatible with browsers that support the HTML5 Canvas.

- CreateJSの応用



紹介する箇所

- 描画 -> EaseLJS
- アニメーション -> TweenJS
- 外部ファイル通信 -> PreloadJS
- 音の制御 → SoundJS
- 応用

描画する

EaseLJSを使用したcanvasの操作

3 STEPS

```
// 1. stageを用意する
const canvasElem = doc.querySelector('.js-canvas');
const stage = new createjs.Stage(canvasElem);

// 2. インスタンスを作成する
const shape = new createjs.Shape();
shape.graphics.beginFill("#FFD09B");
    .drawCircle(0,0,50);

// 3. インスタンスの配置と表示
shape.x = shape.y = 50;
stage.addChild(shape);
stage.update();
```

描画する + イベントを貼る

```
const canvasElemEvent = doc.querySelector('.js-canvas-event')
const stageEvent = new createjs.Stage(canvasElemEvent);

const shapeEvent = new createjs.Shape();
shapeEvent.graphics.beginFill("#FFD09B")
    .drawCircle(0,0,50);

shapeEvent.x = shapeEvent.y = 50;

shapeEvent.addEventListener('click', ()=>{ alert('click'); })
stageEvent.addChild(shapeEvent);
stageEvent.update();
```

描画する + インスタンスを動かす

```
/* Move */
const canvasElemMove = doc.querySelector('.js-canvas-move');
const stageMove      = new createjs.Stage(canvasElemMove);

const shapeMove = new createjs.Shape();
shapeMove.graphics.beginFill('#FFD09B')
    .drawCircle(0,0,50);
shapeMove.x = shapeMove.y = 50;
stageMove.addChild(shapeMove);

createjs.Ticker.setFPS(30);
createjs.Ticker.addEventListener('tick', ()=>{
    shapeMove.x = shapeMove.y += 1;
    stageMove.update();
});
```

RMagic

<https://rmagick.github.io/image1.html#composite>

```
image.composite(image2, Magick::CenterGravity,  
                Magick::OverCompositeOp)
```

アニメーション

TweenJSを使用したアニメーション

get(shape).to(animation)

```
const canvasTween = doc.querySelector('.js-canvas-tween');
const stageTween = new createjs.Stage(canvasTween);

const shapeTween = new createjs.Shape();
shapeTween.graphics.beginFill('#FFD09B')
    .drawCircle(0,0,50);
shapeTween.x = shapeTween.y = 50;
stageTween.addChild(shapeTween);

createjs.Tween.get(shapeTween).to({x: 150, y: 150}, 3000);
createjs.Ticker.addEventListener('tick', ()=>{
    stageTween.update();
});
```

複数アニメーション

```
const canvasTween = doc.querySelector('.js-canvas-tween');
const stageTween = new createjs.Stage(canvasTween);

const shapeTween = new createjs.Shape();
shapeTween.graphics.beginFill('#FFD09B')
    .drawRect(0,0,150, 150);
shapeTween.x = shapeTween.y = 50;
stageTween.addChild(shapeTween);

createjs.Tween.get(shapeTween).to({x: 150, y: 150}, 3000);
createjs.Tween.get(shapeTween, {loop: true}).to({rotation: 360}, 1000);
createjs.Ticker.addEventListener('tick', ()=>{
    stageTween.update();
});
```

連結アニメーション

to / set / wait /call

```
const canvasTween = doc.querySelector('.js-canvas-tween');
const stageTween = new createjs.Stage(canvasTween);

const shapeTween = new createjs.Shape();
shapeTween.graphics.beginFill('#FFD09B')
    .drawRect(0,0,150, 150);
shapeTween.x = shapeTween.y = 50;
stageTween.addChild(shapeTween);

createjs.Tween.get(shapeTween).to({x: 150, y: 150}, 3000)
    .to({rotation: 360}, 1000);
createjs.Ticker.addEventListener('tick', ()=>{
    stageTween.update();
});
```

イージング

http://www.createjs.com/demos/tweenjs/tween_sparktable

```
const canvasTween = doc.querySelector('.js-canvas-tween');
const stageTween = new createjs.Stage(canvasTween);

const shapeTween = new createjs.Shape();
shapeTween.graphics.beginFill('#FFD09B')
    .drawRect(0,0,150, 150);
shapeTween.x = shapeTween.y = 50;
stageTween.addChild(shapeTween);

createjs.Tween.get(shapeTween)
    .to({x: 150, y: 150}, 3000, createjs.Ease.bounceInOut);
createjs.Ticker.addEventListener('tick', ()=>{
    stageTween.update();
});
```

外部ファイル通信

PreloadJSを使って画像の読み込み

画像を描画

* CORSに注意

```
/ * Preload JS */
const canvasPreload = doc.querySelector('.js-canvas-preload')
const stagePreload = new createjs.Stage(canvasPreload);

// 1. LoadQueueインスタンスを作成
const queue = new createjs.LoadQueue();
queue.loadFile('./images/ham-2063533_640.jpg');

queue.addEventListener('fileload', (e)=>{
    const bitmap = new createjs.Bitmap(e.result);
    stagePreload.addChild(bitmap);
    stagePreload.update();
});
```

Canvasの事例

Canvasの注意点

- リソース制限
 - RAMが256MB未満の端末では、3MB以下
 - RAMが256MB以上の端末では、5MB以下
 - <http://lealog.hateblo.jp/entry/2013/03/21/212608>
- canvas size limit
 - IE: height/width: 8,192 pixels
- Retina対応

Canvasからの変換

```
canvas.toDataURL('image/jpeg', 1.0); // image変換  
canvas.toBlob() // Firefox only  
https://developer.mozilla.org/ja/docs/Web/API/HTMLCanvasElement  
antimatter15/jsgif // gif変換
```

おわり