Create.js を使ったデジタル数学教材作成

はじめに

この文書について

Create.jsを使ってsin波等を描画する方法を、JavaScript初学者でもなるべく分かるように記述（ただし、JavaやC言語に関してはある程度の知識がある者を対象）。

What's Create.js?

FlashではAction Scriptと呼ばれるプログラミング言語を使うことができたが、AnimateCCに変更されてからは使用できなくなった。

そこで代替品として生み出されたのがCreate.js。かつてのAction Scriptの文法をそのままに、HTMLとJava Scriptを用いて表現することができる。

（まだ検証していないので推測でしかないが、HTMLとJavaScriptなので、同じページ上でAnimateCCで作ったアニメーションと組み合わせることも可能かもしれない）。

テンプレートファイルの場所

http://54.92.8.231/ の授業一覧にある情報科教育法に、「Create.jsのテンプレート」というのがあるので、ここからダウンロードして使ってください。たいていのPC環境で動作すると思われます。

Create.jsの使い方

基本

loadイベント（画面読み込み）発生時に行う操作を、addEventListener()を使って登録する。このinit関数に、Create.jsを記述していく。

```
<body>
  <canvas id="myCanvas" width="960" height="480"></canvas>

  <script>

    // loadイベント（画面読み込み）発生時に動作させる関数として、init関数を指定する。
    window.addEventListener("load", init);

    function init() {
      // Stageオブジェクトを作成。以後、ここに図を描いていく。
      // idがmyCanvasのcanvasに描かれる。
      var stage = new createjs.Stage("myCanvas");

    }

  </script>
</body>
```

以後のJavaScriptは、すべてinit関数の中に処理を書くこと。

また、**createjs.Stage()**は、引数に与えられたidと一致するcanvasに描画を行うためのオブジェクトを生成する。今回は、myCanvasに対して描画を行うようにする。

円を描画する

以下のように記述することで、startX,startY = (120,200) を中心の座標として、半径r=100の円を描画することができる。

```
// startX, startY にそれぞれ 120, 200 を代入
var startX = 120;
var startY = 200;

// rに100を代入
var r = 100;

// 円のためのシェイプを作る
var circle = new createjs.Shape();
// 線の色を設定する（灰色）
circle.graphics.beginStroke("#888888");
// 中心の座標が(startX,startY)で半径rの円を描く
circle.graphics.drawCircle(startX, startY, r);
// stageに円を置く
stage.addChild(circle);

// 描画を行う。これをしないと描画されないので忘れないこと。
stage.update();
```

new createjs.Shape() と記述することで、**シェイプ**と呼ばれる図形描画のためのオブジェクトを生成することができる。このオブジェクトをcircle変数に格納し、circleの中に入っている同オブジェクトに対し、円を描画するための各種設定を行う。

なお、描画は、**stage.update()**を実行するまで行われない。適切なタイミングで呼び出すことを忘れないこと。

補足として、JavaScirptの変数には**型が存在しない**。代入される値によって、変数の型が動的に決定される。変数を宣言するときは、変数の前にvarと書く（varがないと、多分グローバル変数になる）。

課題1) 小さな円の描画

上で描いた円の中心に、半径が5の小さな円を描画しなさい。

線を描画する

以下のように記述すると、中心をstartX, startYとした、長さ300の水平線が描画される。

```
// 水平線( horizontal line ) のためのシェイプを作る
var hl = new createjs.Shape();
hl.graphics.beginStroke("#888888");
hl.graphics.moveTo(startX-150, startY)
    .lineTo(startX+150, startY);
stage.addChild(hl);
```

これにより、hlのシェイプを使い、moveTo(x, y)とlineTo(x, y)をそれぞれ始点と終点とした線を描画するこ

とができる。

課題2) 垂線の描画

上で描いた線と交差するように、垂線(vertical line)を描画しなさい。

sin波の描画（静止画）

以下のようにすると、0° < θ <　360° のsin波を描画できる。theta変数に対するsin関数は、Math.sin(theta)のように使うことができる。また、円周率は、Math.PIで得ることができる。

```
// サイン波のためのシェイプ
var sinWave = new createjs.Shape();
sinWave.graphics.beginStroke("#FF8888")
  .setStrokeStyle(2) //線の太さを設定
  .moveTo(startX, startY); //始点
stage.addChild(sinWave);

// サイン波を引く（静的）
for ( var i=0; i<360; i++ ) {
  sinWave.graphics.lineTo(startX+i, startY-r*Math.sin(i * Math.PI / 180));
}
```

直線が動くアニメーション

createjs.Ticker.addEventListener()でtickイベントに対し、一定時間ごとに処理を行うように設定することができる。1秒間に60回描画を更新することで、アニメーションが実現できる。

```
var i = 0;
createjs.Ticker.setFPS(60);
createjs.Ticker.addEventListener('tick', function() {
  hl.graphics.lineTo(startX+150+i, startY);
  hl.y = i/5;

  stage.update();
  i++;
});
```

このように記述することで、横に線が伸びつつ下へ動くアニメーションが実現できるはず。

最終課題) sin波を動かす

http://54.92.8.231/wave.htmlのように、sin波が波打つように動かしなさい。