

CurveLib

http://www.libspark.org/wiki/e_s_jp/CurveLib/

Spark project 勉強会 #6
2008/12/19

自己紹介

- TransformationPoint

Flash CS3 でオブジェクトの変形点をパネルから設定する拡張機能。「Apply」ボタンで選択中オブジェクトの変形点を選択モードと入力数値で指定した位置に一括で設定する。

- GetFrameActionScript

.flaファイルのフレームに書かれたActionScriptを調べて、その一覧を出力するコマンド (JSFL) 。

- GetLibraryItemClass

.flaファイルのライブラリでクラス設定されているものを調べてその一覧を出力するコマンド (JSFL) 。

CurveLib

- **CurvePoint.as**
曲線の公式を計算する
半径を1.0として計算
- **ICurveFormula.as**
曲線の公式を定義
半径を1.0として定義

`com.violentcoding.drawing.curvelib.CurvePoint.as`

`com.violentcoding.drawing.curvelib.formula.ICurveFormula.as`

CurvePoint.as

- `getAnglePoint(angle)`
指定角度でのXY座標を取得
- `getPointCount()`
曲線上の計算する点の数を取得
- `startCurvePoint(delay, isClose)`
タイマー計算
- `startTick()/stopTick()/resetTick()`
タイマー計算の制御
- `start/stop/tick/complete/loop` イベント

CurvePoint.as

- **formula**プロパティ
使用する公式 (ICurveFormula)
- **stepSize**プロパティ
何度ごとに計算するか
- **startAngle**プロパティ
開始角度の指定
- **isClockwise**プロパティ
時計周りに計算するかどうか
- **isLoop**プロパティ
ループさせるか

曲線の公式

● Botanic Curve

$$r = 1 + d * \sin(c * \theta)$$

● Rhodonea (バラ曲線)

$$r = \sin(c * \theta)$$

● Epitrochoid (外トロコイド)

$$x = (a + b) \cos(\theta) - h \cdot \cos\left(\left(\frac{a + b}{b}\right) * \theta\right)$$
$$y = (a + b) \sin(\theta) - h \cdot \sin\left(\left(\frac{a + b}{b}\right) * \theta\right)$$

● Hypotrochoid (内トロコイド)

$$x = (a - b) \cos(\theta) + h \cdot \cos\left(\left(\frac{a - b}{b}\right) * \theta\right)$$
$$y = (a - b) \sin(\theta) - h \cdot \sin\left(\left(\frac{a - b}{b}\right) * \theta\right)$$

● Starr Rose

$$x = \left(2 + \sin(a * \theta) / 2\right) * \cos\left(t + \sin(b * \theta) / c\right)$$
$$y = \left(2 + \sin(a * \theta) / 2\right) * \sin\left(t + \sin(b * \theta) / c\right)$$

ICurveFormula

● Botanic Curve

```
new BotanicCurveFormula( cN:uint, dN:uint, cD:uint=1, dD:uint=1 )
```

● Rhodonea (バラ曲線)

```
new RhodoneaFormula( cN:uint, cD:uint=1 )
```

● Epitrochoid (外トロコイド)

```
new EpitrochoidFormula( bN:uint, bD:uint, h:Number=1 )
```

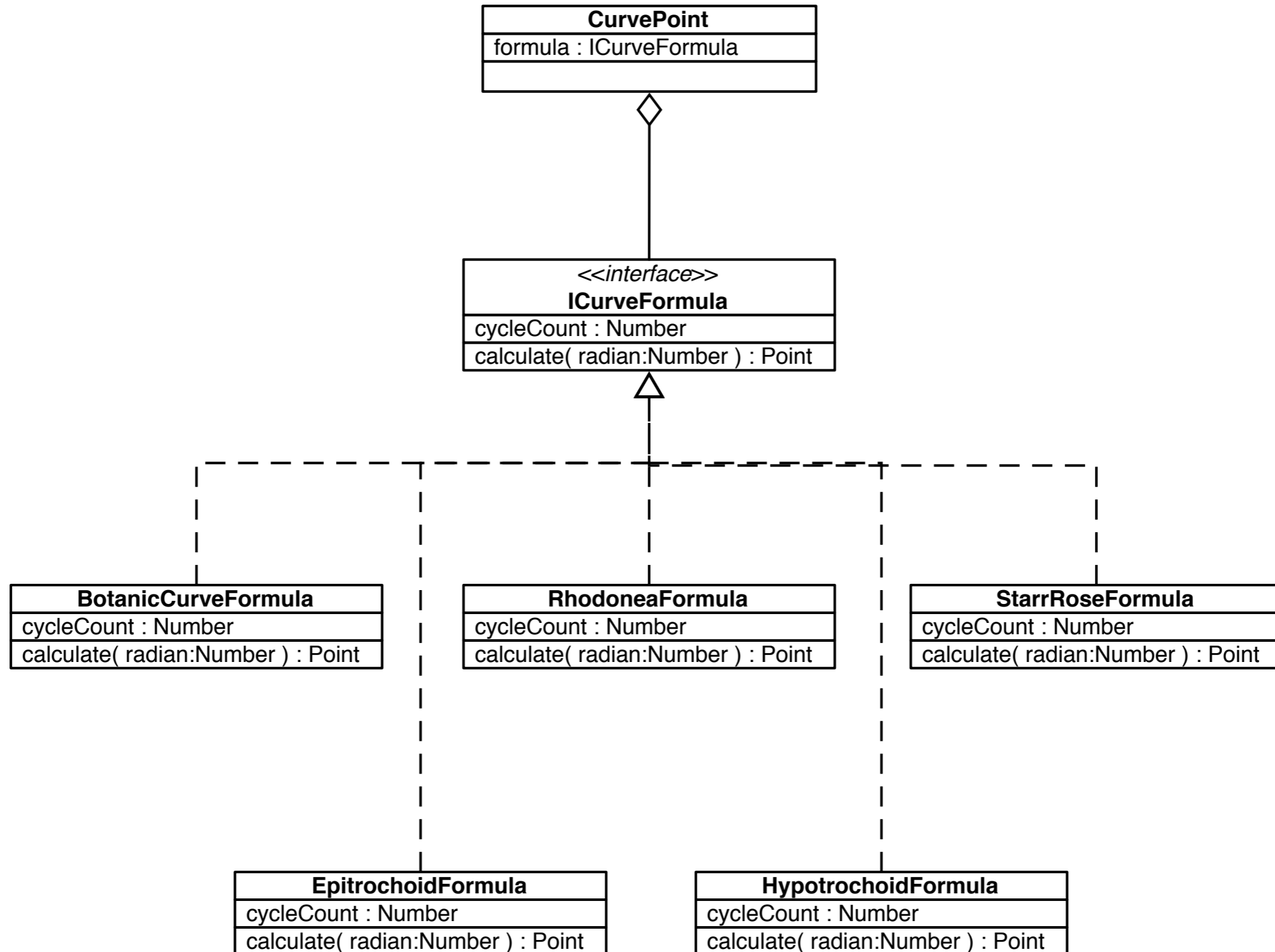
● Hypotrochoid (内トロコイド)

```
new HypotrochoidFormula( bN:uint, bD:uint, h:Number=1 )
```

● Starr Rose

```
new StarrRoseFormula( a:uint, b:uint, c:uint )
```

クラス図



CurveGraphics.as

- CurvePointクラスを使うCurveLibおまけクラス
- CurvePointクラスと「曲線の半径」を指定する
- `graphics.lineTo()`などする
- `for`で一気に描画する方法
- タイマーで描画する方法
- 曲線の中心点が($x=0$, $y=0$)
- `graphics.endFill()`の`true/false`を指定して連続描画

CurveGraphics.as

- `lineCurve(isEndFill)`
線・塗りで曲線を描画
- `chipCurve(chipRadius, isChipFill)`
曲線上の点を円で描画
- `lineAngleBetween(angleBetween)`
曲線上の点Pと指定角度をずらした点Qを結んだ線を描画

trunk/samples/fp9/samples

Sample01.flc · Sample01.as

CurveGraphics.as

- `tickLineCurve(delay, isEndFill)`
線・塗りで曲線をタイマー描画
- `tickChipCurve(delay, chipRadius, isChipFill)`
曲線上の点を円でタイマー描画
- `tickLineAngleBetween(delay, angleBetween)`
曲線上の点Pと指定角度をずらした点Qを結んだ線をタイマー描画

trunk/samples/fp9/samples

Sample02.fla · Sample02.as

CurveGraphics.as

- **isEndFill == true**
線の描画の最後でgraphics.endFill()を実行
次の描画ではあらためてbeginFill()をする
- **isEndFill == false**
線の描画の最後でもgraphics.endFill()をしない
endFill()、beginFill()をしないでそのまま描画

trunk/samples/fp9/samples

Sample03.flc · Sample03.as、 Sample04.flc · Sample04.as

CurveGraphics.as

CurveLib f 01 02 03 04 05 d 01 02 formula

Botanic Curve

$r = 1 + d * \sin(c * \theta)$

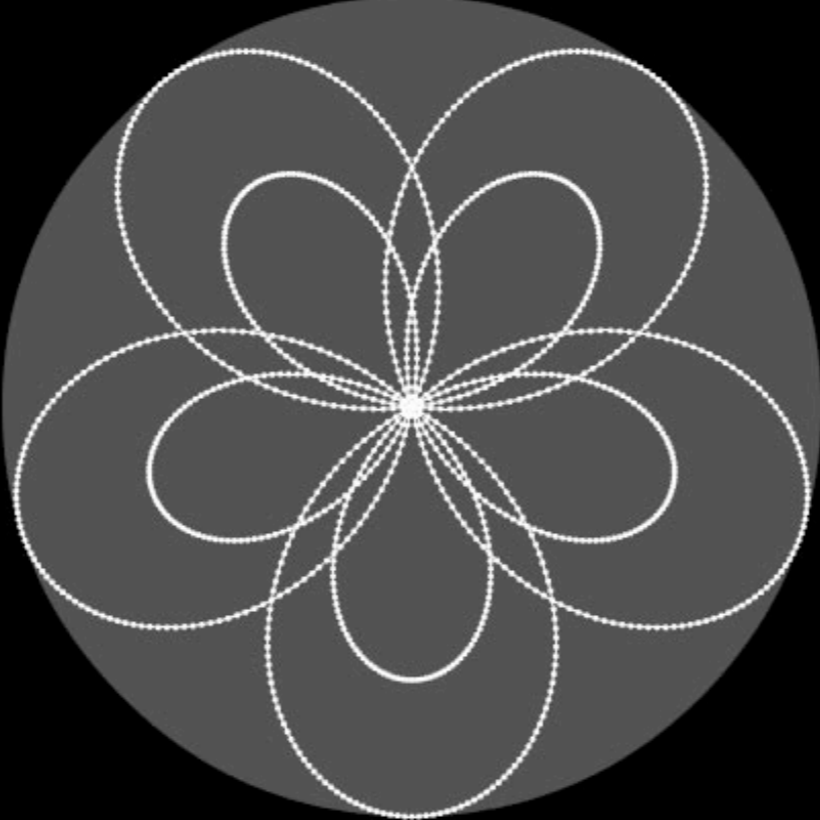
```
new BotanicCurveFormula( cN:uint, dN:uint, cD:uint=1, dD:uint=1 )
```

radius : 200

$c = \frac{5}{3}$ (cN) (cD)

$d = \frac{5}{1}$ (dN) (dD)

- Show Line
- Show Point
- Show Base



tickLineCurve(noFill) ▶

tickLineCurve(Fill) ▶

tickChipCurve ▶

Spark Project/CurveLib ver. 1.0.0

Spark Project/CurveLib ver. 1.0.0

<trunk/doc/fp9/formulaSWF/index.swf>

<http://violentcoding.com/content/curveLib/>