



「物理的な動きを作ってみよう。
enterframeをマスターしよう」

2010.11.27(土)

特別協賛:株式会社かっぺ様

FLASHUP#6

orange-suzuki.com 鈴木克史

本日のメニュー

1. くりかえし処理の基本
2. 雪の動きを作ってみよう
3. ボールの動きを作ってみよう

くりかえし処理の基本

くりかえし処理の作り方

1. くりかえす内容を書いておく
2. くりかえしを実行する

くりかえし処理の書き方

//くりかえす内容

```
function loop(e:Event):void {  
    //-10pxずつ移動させる  
    img1.x -= 10;  
}
```

//くりかえしを実行

```
addEventListener(Event.ENTER_FRAME, loop);
```


くりかえし処理を作ってみよう

やってみよう！

- Lesson1.as
- Lesson2.as

雪の動きを作ってみよう

雪の動きの作り方

1. 雪の粒を作成し配列に入れる
2. `enterFrame`を使って動かす
3. 下まで動かしたら上に戻す

雪の動きを作ってみよう ～その1 オブジェクトの生成～

・ 使うクラス

- ・ Lesson3.as
- ・ SnowView1.as

・ ポイント

クラスを生成して表示させる

雪の動きを作ってみよう

～その2～ 複数のオブジェクトの生成

- ・ 使うクラス

- ・ Lesson3.as

- ・ SnowView2.as

- ・ ポイント

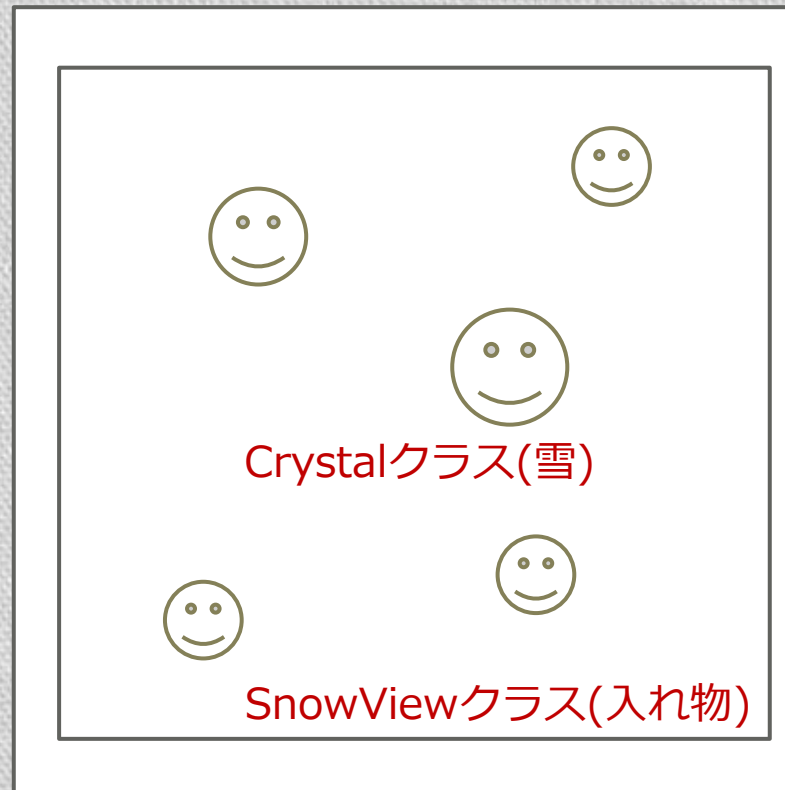
複数のオブジェクトを生成して配列にまとめる

SnowView(入れ物)とCrystal(雪)の関係をつかむ

雪の動きを作ってみよう

～その2～ 複数のオブジェクトの生成

Mainクラス(本体)



雪の動きを作ってみよう ～その3～ 動かしてみる

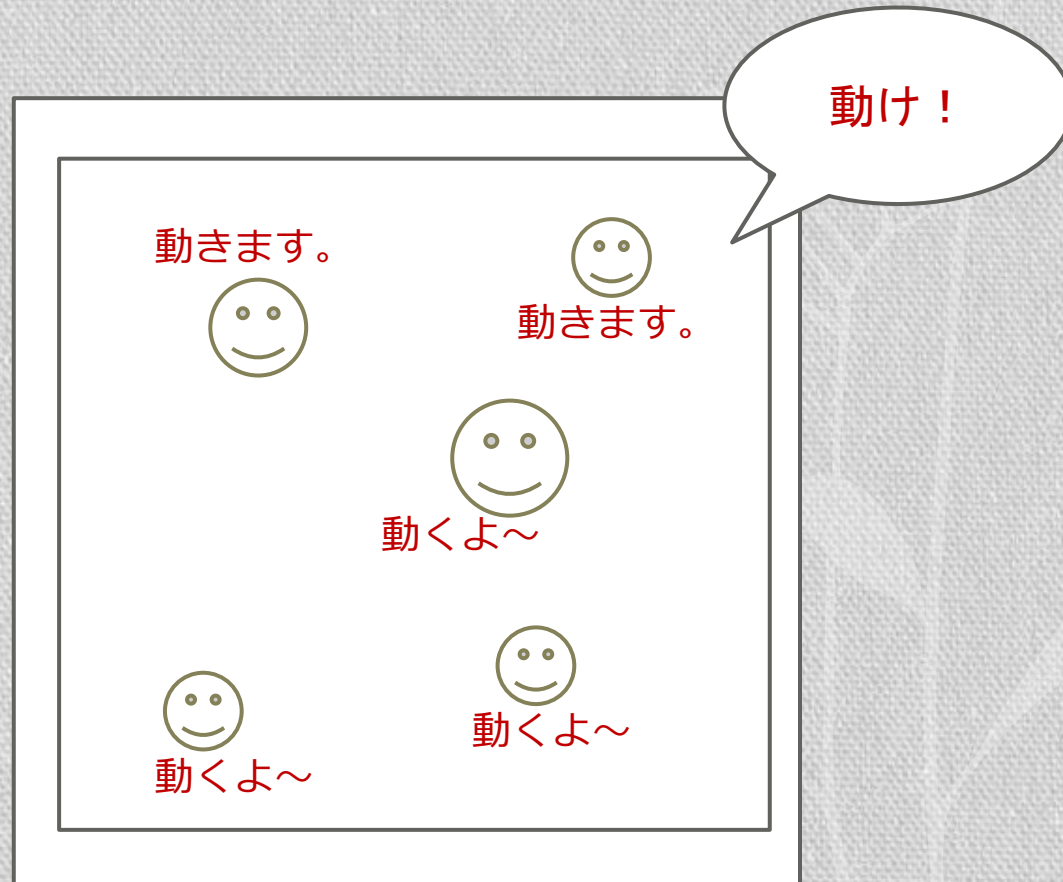
・ 使うクラス

- ・ Lesson3.as
- ・ SnowView3.as

・ ポイント

くりかえしを処理を本体で行い、実際に動かすのは雪のそれぞれに任せる。動くスピードはそれぞれの雪に任せる。

雪の動きを作ってみよう ～その3～ 動かしてみる



雪の動きを作ってみよう

～その4～ 下まで動かしたら上に戻る

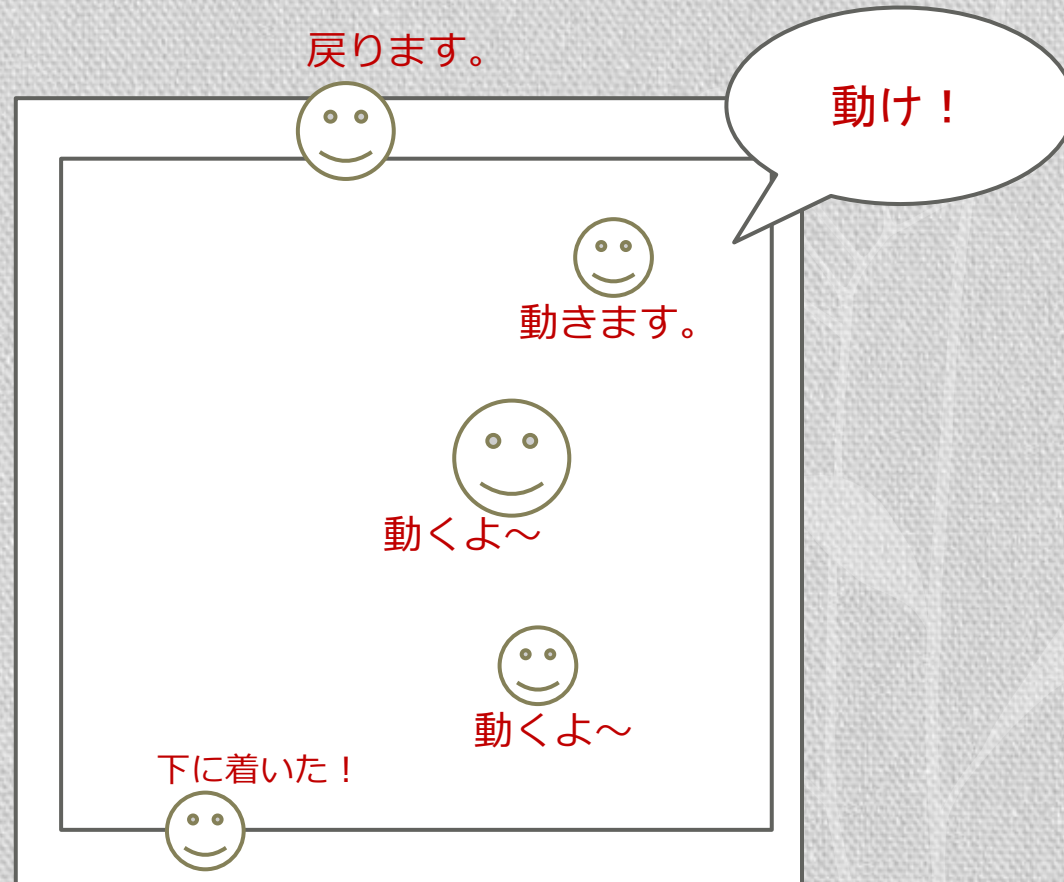
・ 使うクラス

- ・ Lesson3.as
- ・ SnowView4.as

・ ポイント

くりかえし処理のなかでif文を使って判定させる。

雪の動きを作ってみよう ～その3～ 動かしてみる



雪の動きを作ってみよう

～その5～ ウィンドウサイズについて

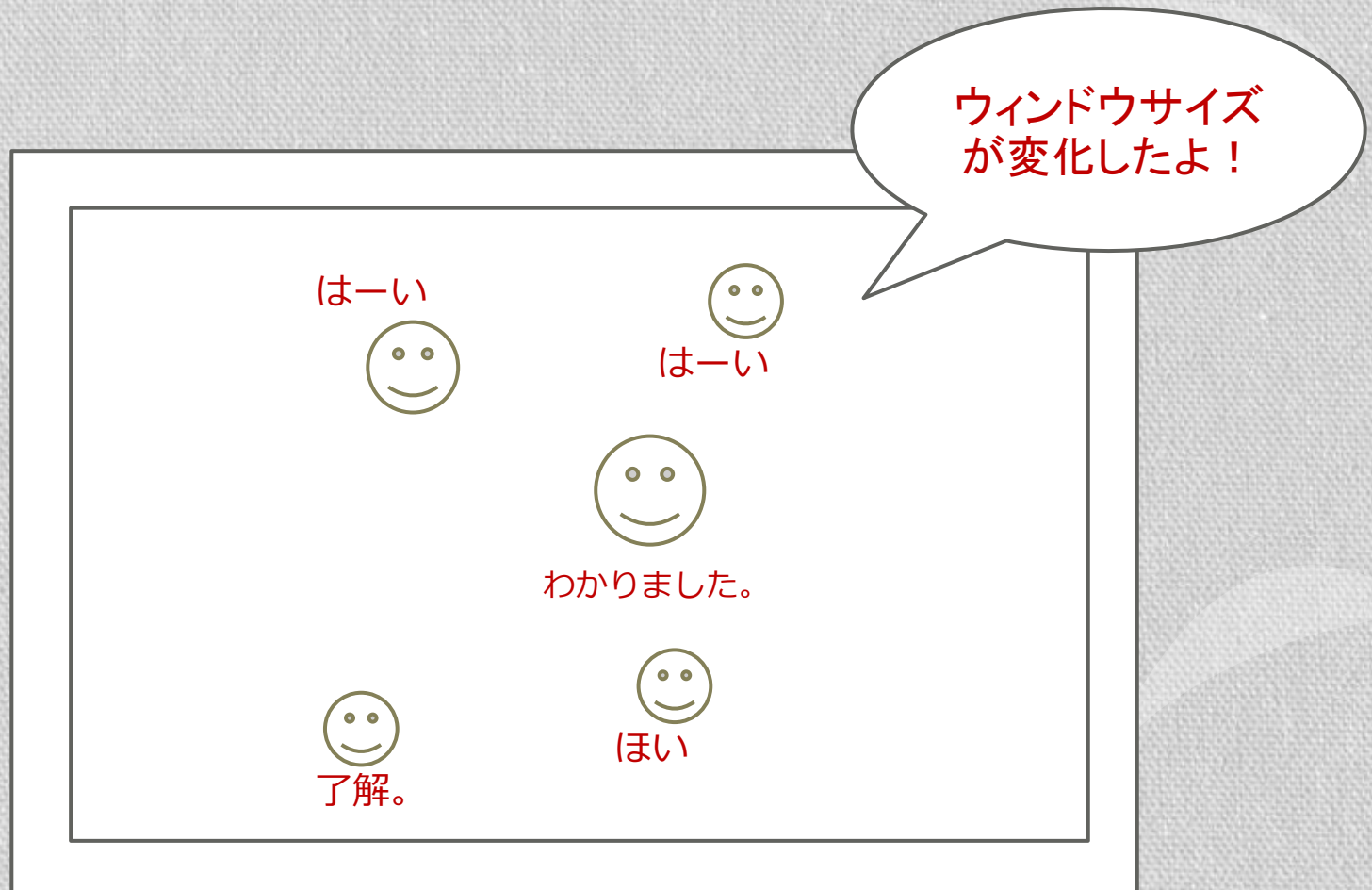
- ・ 使うクラス

- ・ Lesson3.as
- ・ SnowView5.as

- ・ ポイント

ウィンドウサイズが変更したときの処理をつかむ。

雪の動きを作ってみよう ～その3～ 動かしてみる



ボールの動きを作ってみよう

ボールの動きの作り方

1. ボールを作成し配列に入れる
2. enterFrameを使って動かす
3. 壁、天井、床に跳ね返させる
4. ボールの速度の減少、重力を加える
5. マウスに反応させる

ボールの動きを作ってみよう ～その1 オブジェクトの生成～

・使うクラス

- ・ Main.as
- ・ BallView1.as

・ポイント

複数のオブジェクトを生成して配列にまとめる

BallView(入れ物)とBall(ボール)の関係をつかむ

ボールの動きを作ってみよう ～その2 動かしてみる～

・使うクラス

- ・ Main.as
- ・ BallView2.as

・ポイント

くりかえし処理を本体で行い、実際に動かすのは雪のそれぞれに任せる。オブジェクトを生成して配列にまとめる。

ボールの動きを作ってみよう ～その3 はねかえりの処理～

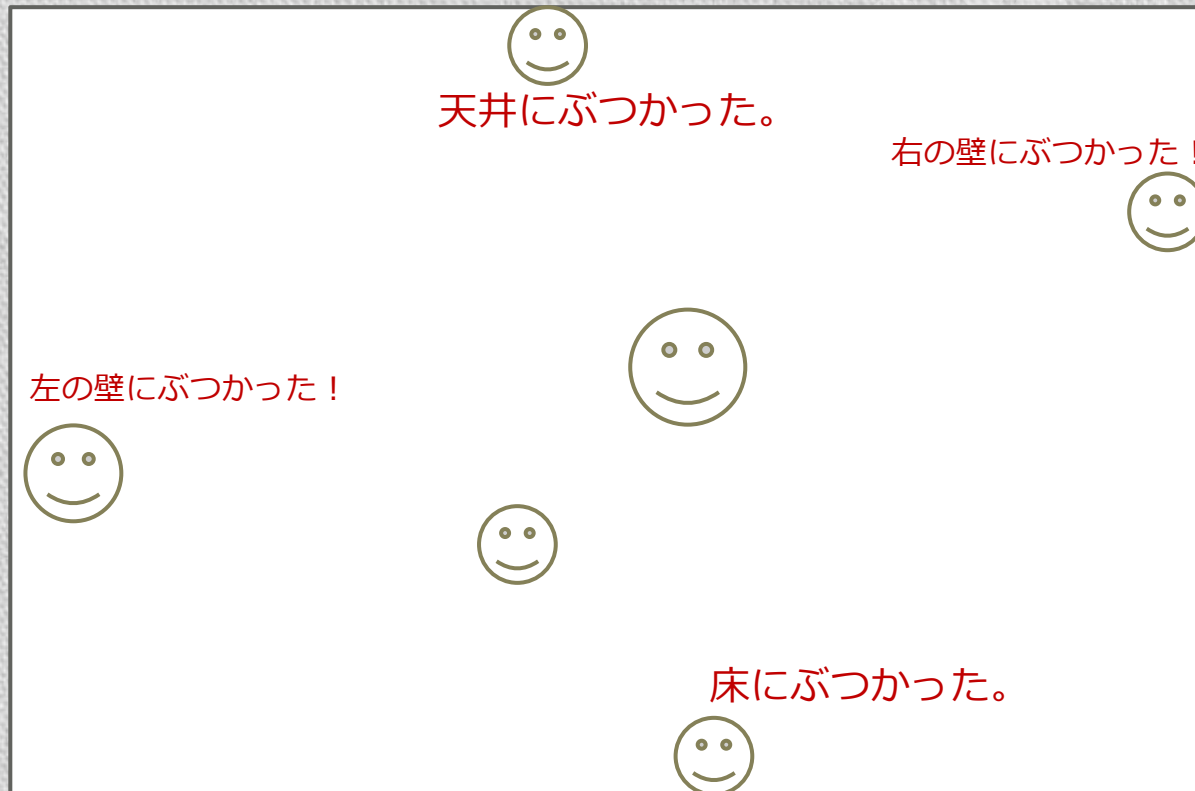
・使うクラス

- ・ Main.as
- ・ BallView3.as

・ポイント

天井、左右の壁、床のそれぞれにぶつかるようにする。

ボールの動きを作ってみよう ～その3 はねかえりの処理～



ボールの動きを作ってみよう ～その4 加速度、重力を加える～

・使うクラス

- ・ Main.as
- ・ BallView4.as

・ポイント

速度の処理と、移動の処理をつかむ。

//減速処理

```
vy = vy *0.98 + 1.6; //減速させ、さらに重力を加える(パターン1)
```

```
(vy += 1.6;) //重力を加える(パターン2)
```

//移動処理

```
y += vy;
```


ボールの動きを作ってみよう ～その5 マウスに反応させる～

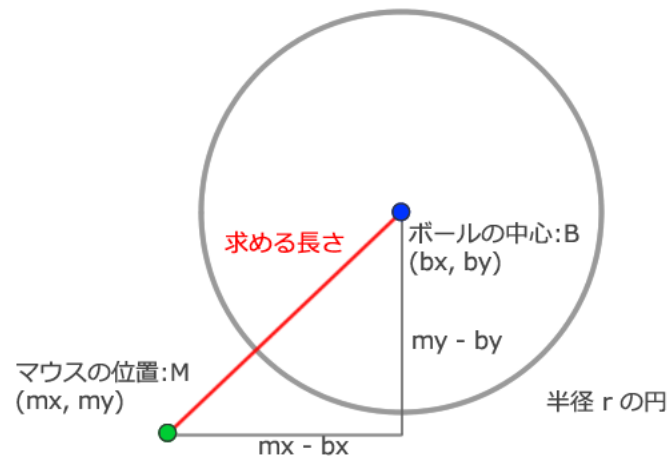
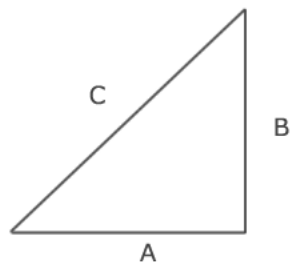
・ 使うクラス

- ・ Main.as
- ・ BallView5.as
- ・ MouseTracer.as

・ ポイント

- ・ マウスとの衝突判定の条件をつかむ。
- ・ マウス速度によって、ボールの勢いを変える。

ボールの動きを作ってみよう ～その6 衝突判定～



三平方の定理

$$C^2 = A^2 + B^2$$



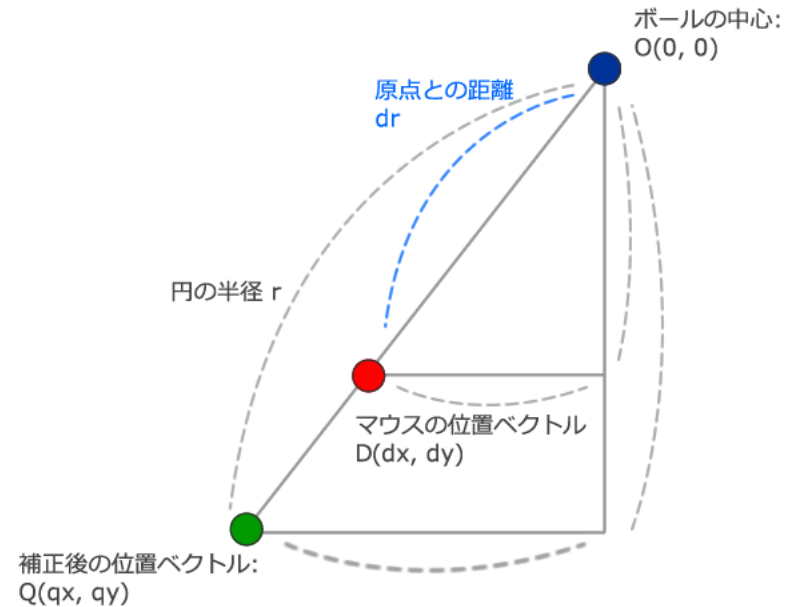
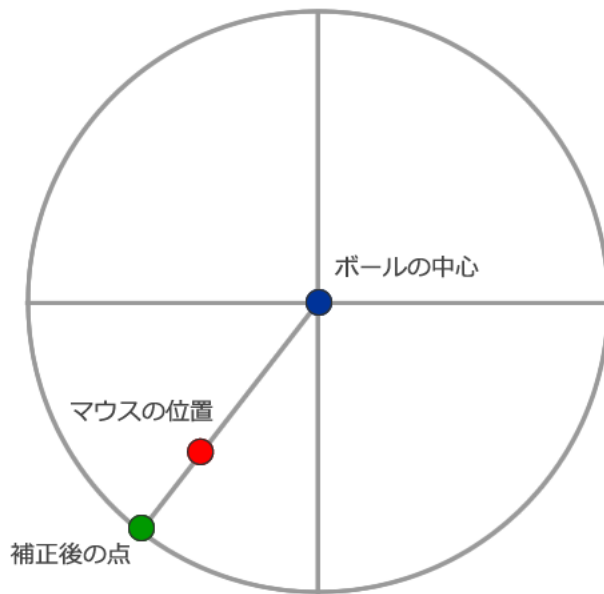
$$C = \sqrt{A^2 + B^2}$$

衝突判定

```
//マウスとボールとの距離  
距離 = Math.sqrt( Math.pow(mx - bx, 2) + Math.pow(my - by, 2) );
```

```
//マウスとボールとの距離がボールの半径を下回ったとき  
if ( 距離 < 半径 ) {  
    衝突した();  
}
```


ボールの動きを作ってみよう ～その7 衝突後の位置補正～



補正後の位置ベクトルの求め方

```
//相似の関係より  
r : dr = qx : dx  
r : dr = qy : dy  
  
qx = dx * r / dr  
qy = dy * r / dr
```

※実際には、入射角を計算した方が正確に求められます。

おわり