今回は１個だけではなく複数のボールが画面の中を飛び回るようにしてみます.

（動画）

等速でころがるボールをたくさん動かしてみる

等速直線運動するボール

今回は説明を簡単にするために次のスケッチをもとに説明をしていきます．これは，等速でころがるボールを表示するスケッチで，前々回に作ったものと機能的には同じです．ただ，時間をつかわず，１回表示するごとに，X 座標，Y 座標に定数を足していくことでボールを移動させています.

３つのボールをころがす

まずは３つのボールをころがしてみましょう．このスケッチは次のようになります.

簡単ですね．ボールの位置と移動量を表す変数を３種類用意し，ボールを動かし，描画し，はねかえりのチェックをするということを３種類の変数にそれぞれについて行ってあげればよいだけです.

先ほどは (x,y) と (dx,dy) という変数名を使いました．今回はわかりやすさのために１つめのボール用には (x0,y0)，(dx0,dy0) とし，２つ目，３つ目はそれぞれ (x1,y1) と (dx1,dy1)，(x2,y2) と (dx2,dy2) としました．なぜ，1 からではなく 0 からにしたかは，後で説明します.

もっとたくさん転がす

もっとたくさん転がすには

ころがすボールを 5 個にするにはどうすればよいでしょうか.

```
ellipse( x3, y3, r, r );
x3 = x3 + vx3;
y3 = y3 + vy3;
if( y3 > height-r || y3 < r ) vy3 = -vy3;
if( x3 > width-r || x3 < r )  vx3 = -vx3;

ellipse( x4, y4, r, r );
x4 = x4 + vx4;
y4 = y4 + vy4;
if( y4 > height-r || y4 < r ) vy4 = -vy4;
if( x4 > width-r || x4 < r )  vx4 = -vx4;
```

を足してあげればよいですよね．もちろん変数の宣言と初期値の代入も忘れないでください.

では，10 個にするにはどうすればよいでしょうか．変数名の後半部分を 5 から 9 としたものを, 同じように書き足していけばよいのですが, 大変ですね．100 個になったら, かなり大変です.

以前，このような同じようなコードを繰り返し書くときには，繰り返し構文を使うということ

を学びました．今回もその手が使えるでしょうか．for 文を使うとすると

```
int i;
for( i = 0 ; i < 10 ; i++ )
{
```

と書いて，次は，

```
  x? = x? + dx?;
```

の ? の部分に i の値が入るようにできればよいのですが，

```
  xi = xi + dxi;
```

のようには残念ながら書けません．では，どうしたらよいでしょうか．

配列の利用

ここで配列というものを使えばうまくいきます．変数は一つの値を入れるための箱のようなものと説明しました．配列はこの箱がつらなったもの，別のたとえを使うと，一つの値を入れられる引き出しを持つタンスのようなものです．そしてその引き出しに値を代入したり，代入されている値を取り出すためにアクセスする際には，配列につけた名前と引出しの番号を指定します．たとえば，

```
x[3] = 10;
y = x[5];
```

のようにです．ここで，引出しのことを正式には要素と呼びます．

配列を使うには，変数と同様にまず宣言をします．宣言は

```
型名　配列名 [ ];
```

と書きます．また，配列を使えるようにするには，

```
配列名 = new 型名 [ 要素数 ];
```

と書きます．これらを同時に，

```
型名　配列名 [ ] = new 型名 [ 要素数 ] ;
```

と書くこともできます．このように書くと，配列名 [0] から配列名 [要素数 -1] までが利用できるようになります．0 から始まる点に注意してください．先のコードで x1 からではなく x0 からにしておいたのは，配列を導入したときと添え字が同じになるようにするためだったのです．

この配列を使えば先の for 文の部分が

```
for( i = 0 ; i < 10 ; i++ )
{
  x[i] = x[i] + dx[i];
```

と書けるようになります.

それではこの配列を使って，10 個のボールを動かすコードを作ってみます．今回は位置（x,y）と移動量（dx,dy) が 10 個分必要なので,

と，すべて 10 個の要素を持った配列として用意すればよいはずです．動かして描画する部分は, i 番目のボールの位置と移動量は，それぞれの配列の i 番目の要素ですので,

とします.

あとは初期値を設定するだけです.

乱数の利用

10 個分のボールの初期位置と移動量を代入する文を書けばよいだけなのですが, なんとなく考えるのも書くのも面倒です．そこで, ここでは乱数を用いて, 適当な値を代入することにします. 乱数とはでたらめ（均一な確率で不規則に）に出現する数です．ただし，コンピュータに作らせる乱数は完全な不規則ではないので疑似乱数と呼ばれています.

乱数を作りたいときには random メソッドを利用します．random メソッドは

```
random();
random(5);
random(3,10);
```

のようにして使います．一番目は 0 以上 1 未満の実数の乱数を生成します．二番目は 0 以上 5 未満の，三番目は 3 以上 10 未満の実数の乱数を生成します．X 座標は r 〜 width-r の値としたいので,

```
x[?] = (int)random( r, width-r );
```

とします．ここで (int) というのは, random メソッドは float 型の値を生成し，それを int 型の変数に格納するため, 型の変換を行うための記述で, キャスト と呼ばれます．int の値の float の値への変換は明示しなくてもよいのですが，float 型の値を int 型の値として利用するときにはきちんとキャストをしないとエラーとなります．なお，float 型から int 型へのキャストは小数点以下は切り捨てられますので，上記のコードは r 以上 width-r 未満の整数が生成されます.

初期値の設定も 10 個のボールについて同じように処理できますので，for 文を使って

と書くことができます．これは最初に 1 回実行すればよいので，setup の中に書けばよいでしょう.

以上をまとめるて，10 個のボールをころがすスケッチを作ってみて下さい.

転がる場所を指定してみる

前回学んだイベントを利用して，クリックをすると，その場所からボールがころがるようにしてみます．

球の数の制御

今回はクリックするまでボールがころがりません．また，クリックするたびにころがるボールの数が増えます．したがって，描画を行っている for 文の条件式のところを

```
for( int i = 0; i < 現在ころがっているボールの数 ; i++ )
```

とする必要があります．そこで，現在ころがっているボールの数を格納するための変数 count を用意することにします．この変数はいろいろなところで使うことになりそうなので，グローバル変数としてコードの先頭部分に記述することにします．

```
int count = 0;
```

この count を用いると，描画の部分は次のようになります．

イベントハンドラの追加

クリックしたときに新しくボールを転がし始めるという処理を行うので，マウスダウン（アップでもよい)イベントのハンドラを用意します．これは前回説明したように mousePressed(または mouseReleased）メソッドを定義することを意味します．

クリックされると, ボールを 1 個ころがすことになるので, 先に用意した変数 count を 1 増やします．ただし, ボールは 10 個までとしているので, if 文を用いて 10 未満だったら増やすという処理にする必要があります．

```
if( count < 10 ) count++;
```

また，ボールを転がしはじめるには，ころがし始める位置を設定する必要もあります．ボールをころがしはじめる位置はクリックした場所なので，クリックされたときに初期化をすることになります．クリックした場所は mouseX と mouseY から得られます．また，移動量の初期化もいっしょにした方がよいでしょう．そこで，setup メソッドの中の初期化コードを消して draw の中でまとめて初期化します．

以上のことから，mousePressed は次のようになります．

なお，count を増やす処理を後ろにおいてあるのには重要な意味があります．count はころがっているボールの数を表しているので, count が 1 のときは, 1 個のボールがころがることになります．そして，そのボールの位置や移動量が格納されている配列の要素番号は 0，つまり count-1 となるため，count の値を増やす前に初期値を代入しているわけです．もし，count を増やす処理を前に持っていった場合は，要素番号の方を count-1 とする必要があります．そうしないと count が 10

のときに, たとえば, x[10] という存在しない要素にアクセスすることになりエラーが発生します.

上記の変更を加えたスケッチを作ってみましょう．なお，前に setup の中に入れていた, x,y,dx,dy の値の初期化は今回は必要ありません．なぜ必要ないかを考えて，削除しましょう．

課題

現在のスケッチでは，すべてのボールが同じ色で表示されています．それぞれのボールが違う色で表示されるようにしてみましょう．それぞれのボールの色を独立に指定するようにし，ボールが転がり始めるときに乱数を用いて色を決めればよいでしょう．なお，色の指定は RGB ではなく, HSB を使った方が, 色相を乱数で決めるだけで, 色を指定することができるので簡単です.

発展課題２

動画のように，ボールの色がどんどん変わっていくようにしてみましょう.

（動画）