

前回までにやったこと

```
float x = glFragCoord.x;  
float y = glFragCoord.y;
```

とすると、座標(ざひょう)を入力できる。

このままでは値が大きいため、グラデーションを書く場合は、小さな数をかけるとちょうどよくなる。

```
float a = 0.01 * x;  
float b = 0.01 * y;
```

この値をgl_FragColorに代入(だいにゆう)すると、グラデーションを書くことができる。

```
gl_FragColor = vec4(a, a, a, 1.0);
```

最小値や最大値の関数を使うと、四角形のグラデーションを書くことができる。

今回やること

(1) 条件分岐をつかって、 $300 < x < 400$ の場所は白 `vec4(1.0, 1.0, 1.0, 1.0)`、そうでない部分を黒 `vec4(0.0, 0.0, 0.0, 1.0)` に塗ろう。

(2) 1 で書いた白い帯(おび)が、時間とともに動くようにしよう。ヒント: 時刻 `time` によって、場所をきめる条件をかえるようにしよう。

(3) `time` の代わりに、`float t = mod(time, 10.0);` を使ってみよう。

(3) 条件を、 $300 < x + y < 400$ などに変えてみよう。

(4) `float a = mod(x, 100.0);` を使ってみよう。

(5) 二種類のアニメーションの結果を足し算してみよう。

```
vec4 p = vec4(a, a, a, 1.0);  
vec4 q = vec4(b, b, b, 1.0);  
gl_FragColor = p + q;
```