

Graphics with Processing



2006-5 色彩とピクセル処理

<http://vilab.org>

塩澤秀和

5.1 色彩

色

- color型
 - 色を表す変数型
- color(成分1, 成分2, 成分3)
 - 色の生成
 - color c = color(r, g, b);
 - 初期モードはRGB & 0~255
- colorMode(色空間, 値範囲)
 - 色指定モードの設定
 - 色空間: RGB, HSB
 - 値範囲: 成分の上限値
 - colorMode(色空間, 範囲1, 範囲2, 範囲3) の形式もある
 - fill(), stroke()などにも影響
 - 例) colorMode(HSB, 1.0);
 - サンプル Examples → Color

色の混合

- 透明度(alpha)
 - 色の第4成分
 - 重ね塗りでの濃さ
 - c = color(r, g, b, a);
 - fill()やstroke()でも指定可
 - 例) fill(255, 0, 0, 128);
- blend(色1, 色2, 混色演算)
 - 色の混合
 - 混色演算: BLEND, ADDなど
 - 新バージョンでは blendColor

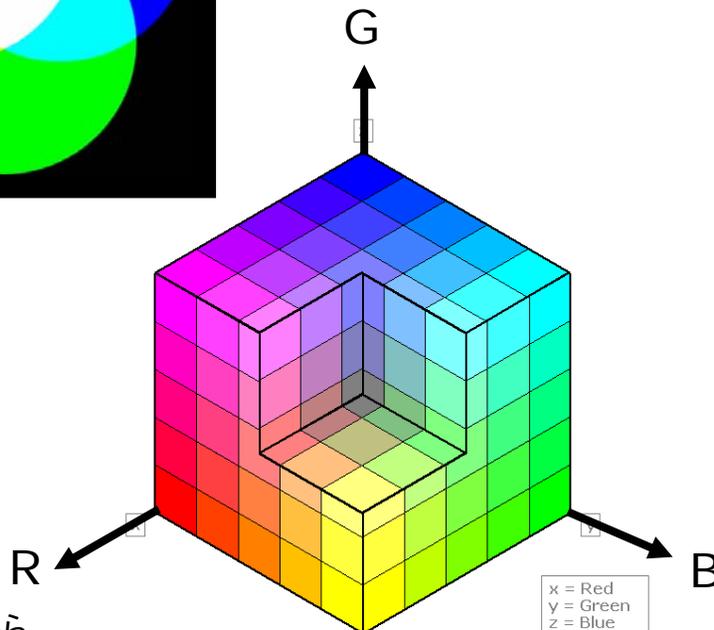
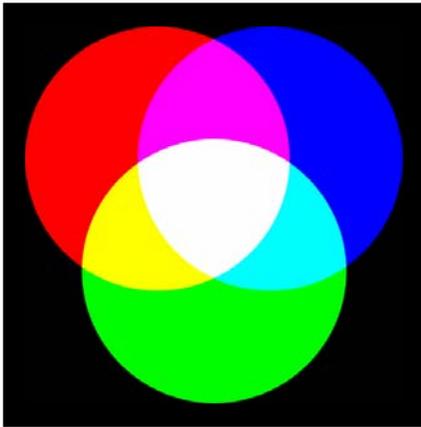
色の成分の取得

- red(c), green(c), blue(c), hue(c), saturation(c), brightness(c), alpha(c)

5.2 色空間

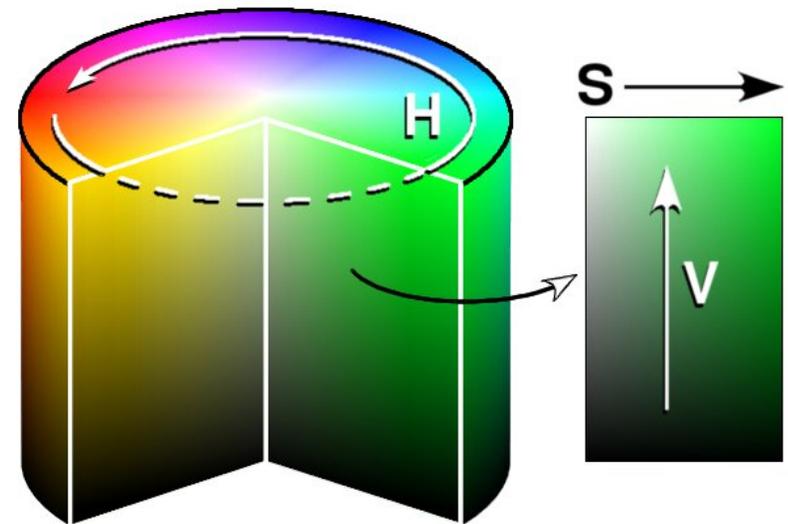
RGB色空間

- 光の三原色(赤, 緑, 青)



HSB(HSV)色空間

- 色相(H):色あい
- 彩度(S):あざやかさ
- 明度(B or V):明るさ



5.3 画像

画像

- PImage型
 - 画像を表す変数型
 - サンプル File → Examples → Image → Sprite
- loadImage("ファイル名")
 - 画像の読み込み
 - 通常, setup()の中で準備
 - 対応形式 .gif .jpg .png .tga
 - Sketch → Add File...で, あらかじめ, 画像ファイルをデータフォルダにコピーしておくこと
- get(), get(x, y, 幅, 高さ)
 - 画面領域を画像として取得
 - PImage img = get();

画像表示

- image(画像, x, y)
 - 画像の描画
- image(画像, x, y, 幅, 高さ)
 - サイズを変更して画像を描画
- imageMode(モード)
 - rectMode/ellipseModeと同様

画像の部分表示

- copy(画像, x_{画像}, y_{画像}, w_{画像}, h_{画像}, x, y, w, h)
- blend(画像, x_{画像}, y_{画像}, w_{画像}, h_{画像}, x, y, w, h, 混色演算)
 - 画像の指定領域を描画

5.4 ピクセル処理

ピクセル処理

- pixels[]
 - color型配列
 - グローバル変数
 - 画面座標(x, y)の画素の色
pixels[y * width + x]
- loadPixels()
 - ピクセル処理の準備
 - 画面の画素ごとの色データを
pixels[]に読み込む
- updatePixels()
 - ピクセル処理の終了
 - pixels[]を画面に反映する
- ピクセル操作
 - c = pixels[y * width + x];
 - pixels[y * width + x] = c;

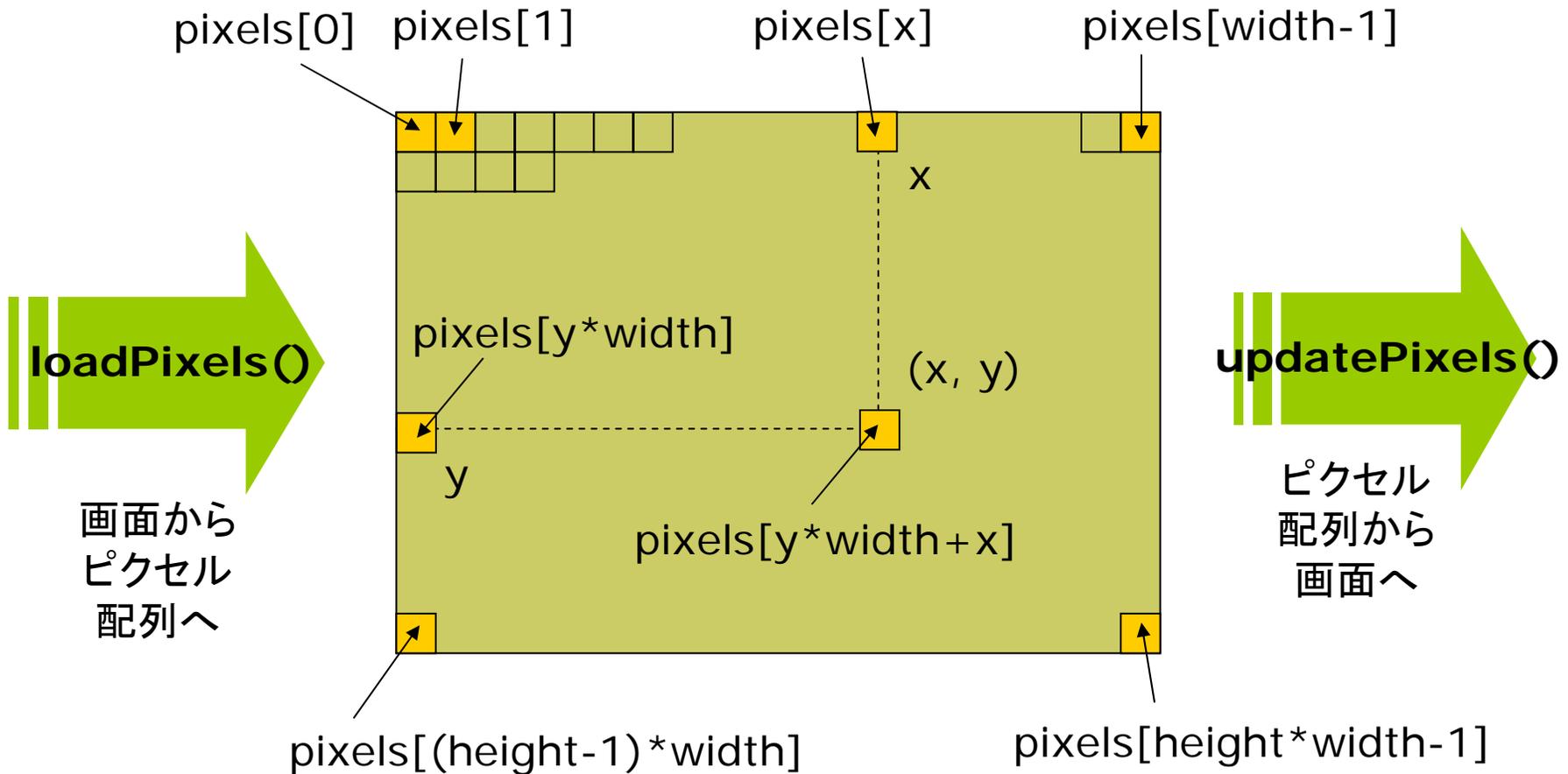
画面領域のコピー

- copy(x1, y1, w1, h1, x2, y2, w2, h2)
- blend(x1, y1, w1, h1, x2, y2, w2, h2, 混色演算)
 - 画面領域を別の場所にコピー
 - 混色演算はマニュアル参照

画像のピクセル処理

- PImageのフィールドとメソッド
 - 画像の変数名を img とすると
 - img.width, img.height
 - img.pixels[]
 - img.get(x, y, 幅, 高さ)
- サンプル Examples → Image
→ Brightness

5.5 ピクセル配列



5.6 演習課題

準備

- 右のプログラムに, 適当なsetup()関数を補いなさい
- プログラムを実行したら, ウィンドウ上でマウスをドラッグしてみなさい
- さらに, 3行目forの内側に
if (random(1.0) < 0.2) { ...
という条件判定を入れてみなさい

課題

- 上から下に流れている模様を, 右から左に流れるように変更しなさい
- さらに今回の講義内容を使って自由なプログラムを作成してみなさい
- 提出URL: <http://vilab.org/upload/cg-upload.html>
- 画像を使う場合はプログラムのフォルダをまとめてzipに圧縮して提出

```
void draw() {  
  loadPixels();  
  for (int x = 0; x < width; x++) {  
    for (int y = height-1; y > 0; y--) {  
      pixels[x + y*width] =  
        pixels[x + (y-1)*width];  
    }  
    pixels[x] = color(0, 0,  
      frameCount % 256);  
  }  
  updatePixels();  
  if (mousePressed) {  
    noStroke();  
    fill(255, 220, 220, 200);  
    ellipse(mouseX, mouseY, 20, 20);  
  }  
}
```