



プログラミング演習(12)

マルチメディア

中村, 高橋
小林, 橋本



- Processing で画像や音楽を扱う
 - 画像を表示する
 - 音楽を再生する
 - 効果音を再生する

画像の表示



PImage 画像用変数;

画像用変数 = loadImage("画像名"); で準備

image(画像用変数, x座標, y座標); で表示する

画像はプログラムにドロップで利用可能に

(ドラッグアンドドロップしないと使えない)

```
PImage mapImage = loadImage("map.png");  
size(640, 400);  
background(255, 255, 255);  
image(mapImage, 0, 0);
```



- 場所だけを指定した画像の描画
 - `image(画像用変数, X座標, Y座標);`
 - サイズは画像自体の大きさになる
- 場所とサイズを指定した画像の描画
 - `image(画像用変数, X座標, Y座標, 横幅, 縦幅);`
 - 拡大縮小やゆがめた描画も可能

カーソルの場所に画像を表示



(Q) 画像をカーソルの下に常に表示するには
どうするか？

• 考え方

- 画像を用意して、プログラムにドロップ！
- PImage 型の変数を作成 (cursorImage など)
- loadImage で画像を読み込む
 - cursorImage = loadImage("gazo.jpg");
- draw の度に背景を塗りつぶす
- draw の度に image でマウス位置に画像を表示
 - image(cursorImage, mouseX, mouseY);

カーソルの場所に画像を表示

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



```
PImage cursorImage;

void setup()
{
  size(800, 600);
  cursorImage = loadImage("gazo.jpg");
}

void draw()
{
  background(255);
  image(cursorImage, mouseX, mouseY);
}
```

パラパラアニメーション

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



(Q) 用意した10枚の画像をパラパラ切り替えるアニメーションを作りたい



- 考え方

- 画像を10枚用意（ペイントで描いても，写真を撮影してもOK．名前は適当に順番を付けましょう）
- 要素数が10の PImage 型の配列を作る
 - PImage [] parapara = new PImage [10];
- setup で画像をすべて読み込む（loadImage）
 - parapara[0] = loadImage("gazo0.jpg");
- draw で表示する画像番号を変数 i として準備
- draw の度に i 番目の画像を表示
- i が10になったら0に戻す
- frameRate(10); で draw の更新速度を設定

パラパラアニメーション

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



```
PImage [] parapara = new PImage [10];
int i = 0;

void setup()
{
  size(800, 600);
  parapara[0] = loadImage("gazo0.jpg");
  parapara[1] = loadImage("gazo1.jpg");
  parapara[2] = loadImage("gazo2.jpg");
  parapara[3] = loadImage("gazo3.jpg");
  parapara[4] = loadImage("gazo4.jpg");
  parapara[5] = loadImage("gazo5.jpg");
  parapara[6] = loadImage("gazo6.jpg");
  parapara[7] = loadImage("gazo7.jpg");
  parapara[8] = loadImage("gazo8.jpg");
  parapara[9] = loadImage("gazo9.jpg");
  frameRate(10);
}
```

```
void draw()
{
  image(parapara[i], 0, 0);
  i++;
  if(i==10)
  {
    i=0;
  }
}
```

ちなみに



```
parapara[0] = loadImage("gazo0.jpg");  
parapara[1] = loadImage("gazo1.jpg");  
parapara[2] = loadImage("gazo2.jpg");  
parapara[3] = loadImage("gazo3.jpg");  
parapara[4] = loadImage("gazo4.jpg");  
:  
parapara[9] = loadImage("gazo9.jpg");
```




```
for( int j = 0; j < 10; j++)  
{  
    parapara[j] = loadImage("gazo" + j + ".jpg");  
}
```

ちなみに



```
parapara[0] = loadImage("gazo0.jpg");  
parapara[1] = loadImage("gazo1.jpg");  
parapara[2] = loadImage("gazo2.jpg");  
parapara[3] = loadImage("gazo3.jpg");  
parapara[4] = loadImage("gazo4.jpg");  
:  
parapara[9] = loadImage("gazo9.jpg");
```



```
int j = 0;  
while(j < 10)  
{  
    parapara[j] = loadImage("gazo" + j + ".jpg");  
    j++;  
}
```

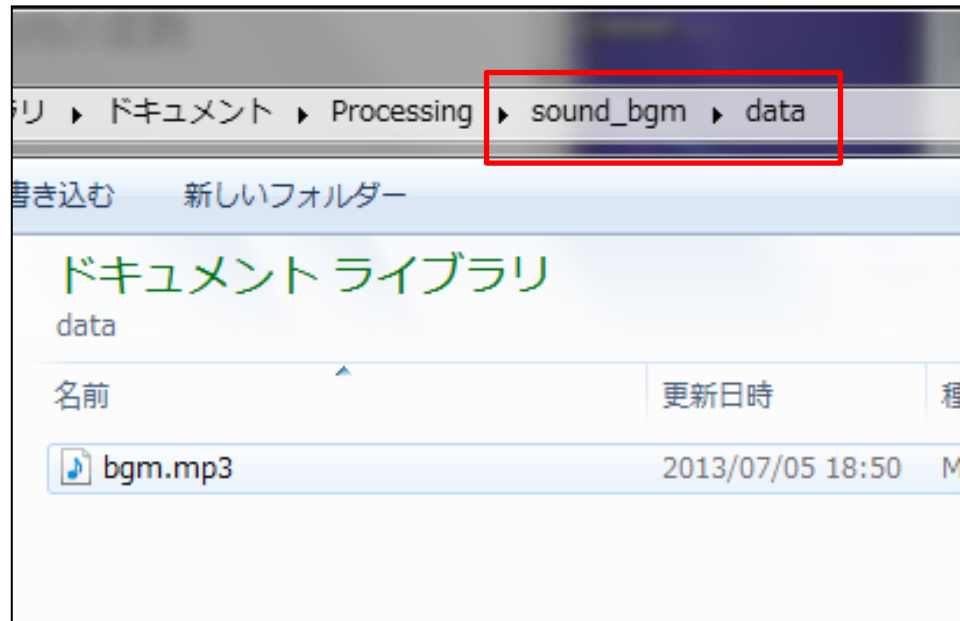
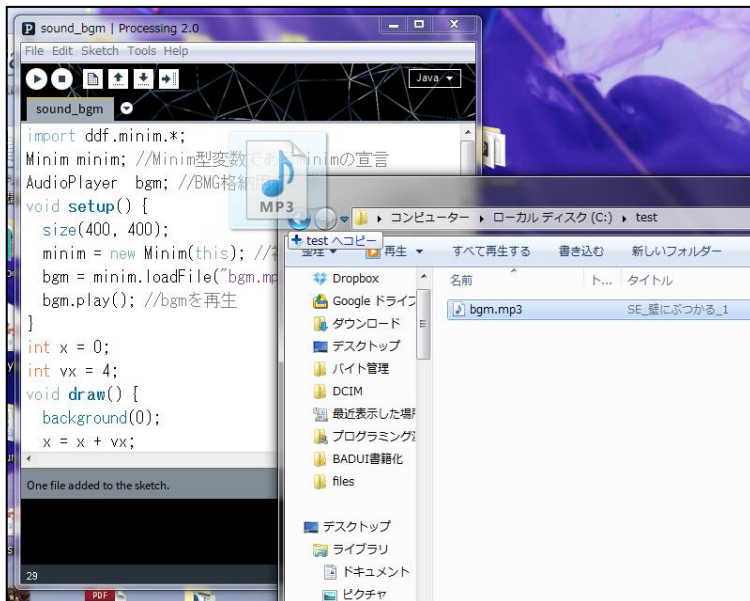


- ある程度小さな画像が，画面の端で衝突して跳ね返るプログラムを作ろう
 - 画像の x , y 座標と速度 v_x , v_y を用意し，`draw` の度に座標を変更し，端で跳ね返る
- 画像を利用して，占いをするプログラムを作ろう
 - 大吉，中吉，小吉，大凶の画像を用意する
- マウスカーソルの後を10個の画像が追尾するプログラムを作ろう

音楽の再生



- minim と AudioPlayer を利用
- 「準備と再生」「終了」が必須
- Processing の画面にファイルをドラッグアンドドロップする！（dataフォルダに保存される）





- 準備：グローバル & setup()
 - 音楽再生ライブラリ（便利関数群）の読み込み
 - `import ddf.minim.*;`
 - Minimの変数を初期化し，初期化
 - Minim はサウンド関係を扱うクラス
 - `Minim minim; // をグローバル変数として用意`
 - `minim.loadFile("ファイル名");` でファイルを読み込む
 - AudioPlayer を初期化し loadFile の結果を受け取る
 - AudioPlayer は音声 / 音楽の再生を司るクラス
 - `AudioPlayer bgm = minim.loadFile("ファイル名");`
 - `bgm.loop(); // 繰り返し再生`
 - `bgm.stop(); // 停止`



- 終了: `stop()`
 - `void stop(){ ... }` は, `void setup(){ ... }` が最初に呼びだされるように, 最後に呼びだされる関数
 - `AudioPlayer` の終了
 - `bgm.close();`
 - `Minim` の終了
 - `minim.stop();`
 - 親の終了 (`stop()` の中では最後に必ず書く)
 - `super.stop();`
 - 今はおまじないだと思っておいて下さい

音楽の再生



```
import ddf.minim.*;
Minim minim; //Minim型変数であるminimの宣言
AudioPlayer bgm; //BGM格納用の変数
int x = 0;
int vx = 4;

void setup()
{
  size(400, 400);
  minim = new Minim(this); //初期化
  bgm = minim.loadFile("bgm.mp3"); //mp3ファイルを指定する
  bgm.play(); //bgmを再生
}

void draw()
{
  background(0, 0, 0);
  x = x + vx;
  if(x >= width)
  {
    x = width;
    vx = -vx;
  }
  else if(x <= 0)
  {
    x = 0;
    vx = -vx;
  }
  ellipse(x, 200, 20, 20);
}

void stop()
{
  bgm.close(); //サウンドデータを終了
  minim.stop();
  super.stop(); // 必須
}
```




- 準備: グローバル & setup()
 - 音楽再生ライブラリ (便利関数群) の読み込み
 - `import ddf.minim.*;`
 - Minim の変数を定義し, 初期化
 - AudioSnippet を初期化し `minim.loadSnippet` の結果を受け取る
 - AudioSnippet は音声の再生を司るクラス
 - AudioSnippet crash
 - `= minim.loadSnippet("ファイル名");`
 - crash は変数名. 他の名前でもOK



- 再生処理：
 - `crash.play()`; // `crash`に格納された音の再生
 - `crash.rewind()`; // `crash`に格納された音の巻き戻し
- 終了: `stop()`
 - `void stop(){ ... }` は, `void setup(){ ... }` が最初に呼びだされるように, 最後に呼びだされる関数
 - `AudioSnippet` の終了
 - `crash.close()`;
 - `Minim` の終了
 - `minim.stop()`;
 - 親クラスの終了 (`stop()` の中では最後に必ず書く)
 - `super.stop()`;

効果音の再

```
import ddf.minim.*;
Minim minim; //Minim型変数であるminimの宣言
AudioSnippet crash; //衝突サウンド格納用の変数
int x = 0;
int vx = 4;
void setup()
{
  size(400, 400);
  minim = new Minim(this); //初期化
  crash = minim.loadSnippet("crash.mp3"); //mp3ファイルを指定する
}
void draw()
{
  background(0, 0, 0);
  x = x + vx;
  if(x >= width)
  {
    x = width;
    vx = -vx;
    crash.rewind();
    crash.play(); //再生
  }
  else if(x <= 0)
  {
    x = 0;
    vx = -vx;
    crash.rewind();
    crash.play(); //再生
  }
  ellipse(x, 200, 20, 20);
}
void stop()
{
  crash.close(); //サウンドデータを終了
  minim.stop();
  super.stop(); // 必須
}
```

予習問題



- 2つの移動する円を用意し，円が壁に衝突する度に衝突音が鳴るようにせよ
- 2つの移動する円を用意し，円が壁に衝突する度にそれぞれ違う音が鳴るようにせよ
- 2つの移動する円について，マウスでクリックすると破裂音が鳴るようにせよ