

情報メディア 基盤ユニット

サウンド情報の取り扱い

情報メディア学科佐藤尚

最終課題制作のお知らせ

提出物

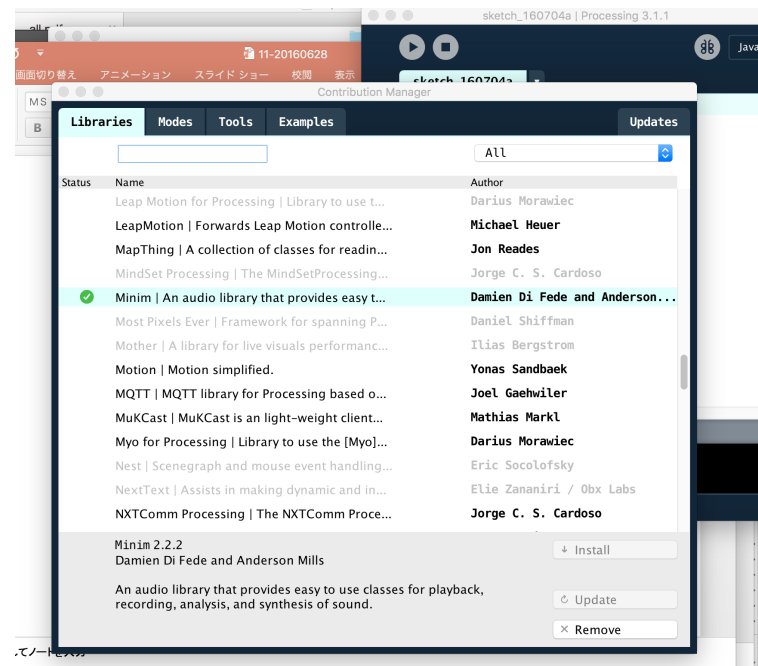
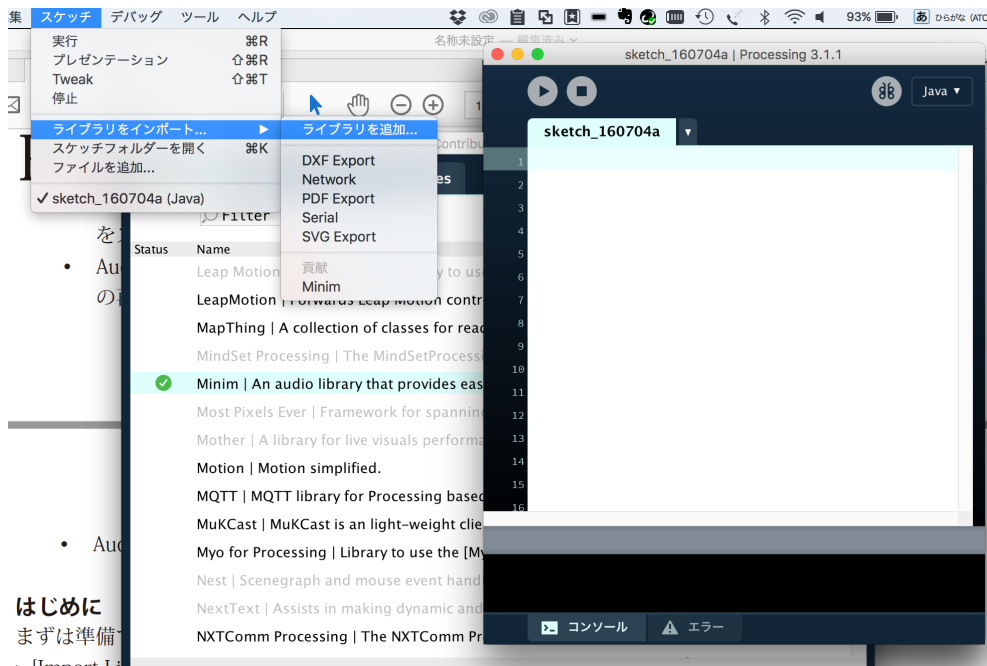
- Processingで作ったプログラム
 - （「スケッチをアーカイブ」の機能を利用して、zipファイルとして提出）
- 自分の制作物の内容を説明するWordで作ったレポート
- 5分間のプレゼンテーションを想定した、自分の作品紹介用のPowerPoint

提出締め切り：7月26日（火）の2限授業終了時まで

発表会：7月29日（金）の演習時

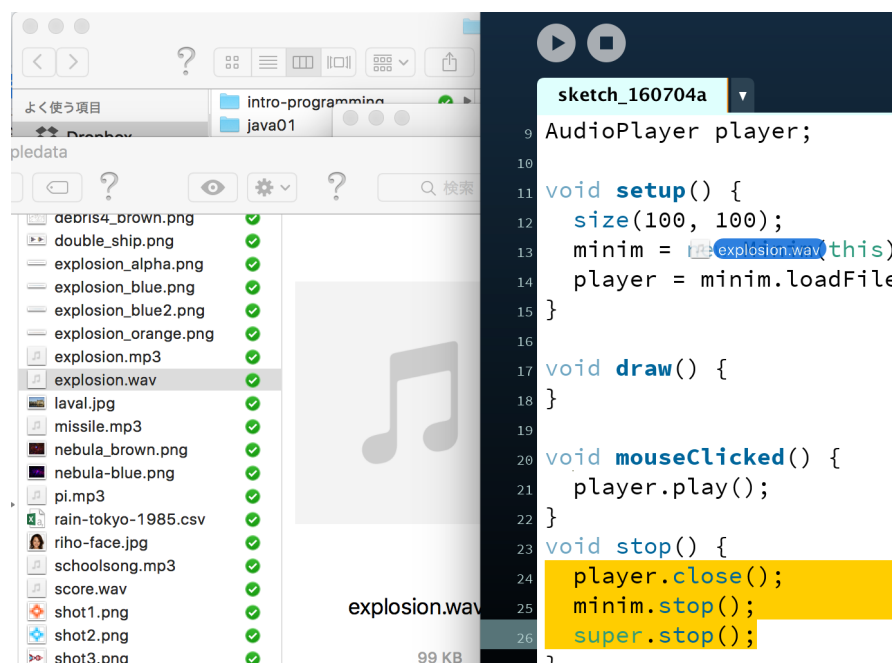
サウンドライブラリを 使うために

サウンドライブラリMinimのインストールが必要



サウンドファイルの置き場

画像ファイルなどと同じように
スケッチフォルダ内のdataフォルダに置く



Processingでの サウンドファイルの再生

AudioPlayer

ファイルサイズの大きいサウンドファイルを再生します。

AudioSnippet

ファイルサイズの大きくサウンドファイルを再生します。

AudioSample

ファイルサイズの小さなサウンドファイルを再生します。

再生できる サウンドファイル形式

WAV

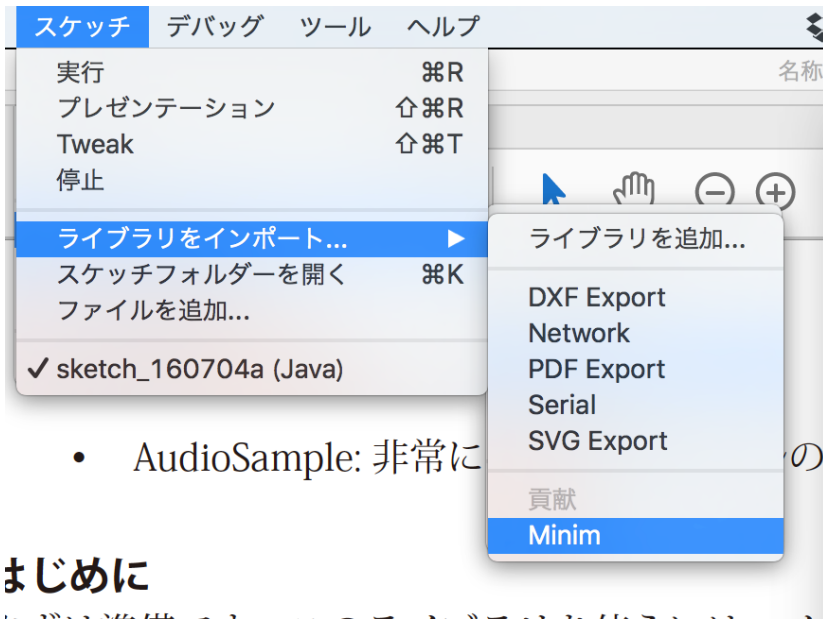
AIFF

AU

MP3

SND

サウンドライブラリの インポート



The screenshot shows the Processing IDE code editor with the following code:

```
sketch_160704a  
1 import ddf.minim.*;  
2 import ddf.minim.analysis.*;  
3 import ddf.minim.effects.*;  
4 import ddf.minim.signals.*;  
5 import ddf.minim.spi.*;  
6 import ddf.minim.ugens.*;  
7
```

大きめサイズの サウンドファイルの再生

AudioPlayer型を利用する

Minimオブジェクトの生成

```
minim = new Minim(this);
```

サウンドファイルの読み込み

```
player =  
minim.loadFile(ファイル);
```

サウンドファイルの再生

```
player.play();  
player.loop();
```

後始末

```
player.close();  
minim.stop();  
super.stop();
```


Minimオブジェクトの生成

Processingの中でサウンドの機能を制御するプログラムを作る。



Minim型の変数`minim`に保存しておく
setup関数内で作る

```
minim = new Minim(this);
```

変数`minim`はMinim型の大域変数
`Minim minim;`

サウンドファイルの読み込み

```
player = minim.loadFile("ファイル名");
```

AudioPlayer型の大域変数player

サウンド再生の制御

play()

play(millis)

loop()

pause()

rewind()

close()

後始末

そのサウンドを
再生しない

```
player.close();
```

サウンドの
機能を使わない

```
minim.stop();    stop関数内で実  
super.stop();   行するのが一般的
```

stop関数はプログラム終了時に自動的に呼びだされる

余り大きくくないサイズの サウンドファイルの再生

AudioSnippet型を利用する

Minimオブジェクトの生成

```
minim = new Minim(this);
```

サウンドファイルの読み込み

```
player =  
minim.loadSnippet(ファイル);
```

サウンドファイルの再生

```
player.play();
```

後始末

```
player.close();  
minim.stop();  
super.stop();
```

サウンドファイルの読み込み

```
player = minim.loadSnippet("ファイル名");
```

AudioSnippet型の大域変数player

小さいサイズの再生 効果音などに利用

AudioSample型を利用する

Minimオブジェクトの生成

```
minim = new Minim(this);
```

サウンドファイルの読み込み

```
player =  
minim.loadSample(ファイル);
```

サウンドファイルの再生

```
player.trigger();
```

後始末

```
player.close();  
minim.stop();  
super.stop();
```

サウンドファイルの読み込みと再生

```
player = minim.loadSample("ファイル名");
```

AudioSample型の大域変数player

AudioSample型の再生にはtriggerを使う

```
player.trigger();
```


再生以外に出来ること

メタデータの取得

11章の10ページの表11-4を参照
サンプルプログラム11-7

再生時のデータ取得

length() //全体の演奏時間 (ミリ秒)
position()//再生経過時間 (ミリ秒)
isPlaying()//再生中かどうか？

正弦波などの再生

Minimオブジェクトなどの生成

```
minim = new Minim(this);  
out =  
minim.getLineOut(Minim.STEREO);
```

波形データの生成

```
sine =  
new SineWave(440, 0.5,  
out.sampleRate());
```

波形データの再生

```
out.addSignal(sine);
```

後始末

```
out.close();  
minim.stop();  
super.stop();
```

Minimオブジェクトなどの生成

```
minim = new Minim(this);
```

```
out = minim.getLineOut(Minim.STEREO);
```

AudioOutput型の大域変数`out`

波形データの生成

```
sine =  
new SineWave(440, 0.5, out.sampleRate());
```

SineWave型の変数に保存

生成できる基本波形データ

正弦波

SineWave(周波数,振幅,サンプリングレート)

矩形波

SquareWave(周波数,振幅,サンプリングレート)

ノコギリ波

SawWave(周波数,振幅,サンプリングレート)

振幅は0~1の数値、
サンプリングレートはsampleRate() で取得すればよい

生成できる波形データ

addSignal() で複数の基本波形データをミックスして発生させることができる

正弦波

矩形波

ノコギリ波

メソッド	機能
setFreq(周波数)	派生させる音の周波数を変更する
setAmp(振幅)	派生させる音の振幅を変更する(0~1)
setPan(パン位置)	-1 (左チャンネルのみ) ~ 0 (中央) ~ 1 (右チャンネルのみ) の範囲の数値でパン位置を設定する

16ページのサンプル11-14とか

マイクからの サウンドデータの取得

AudioInput型を利用する

Minimオブジェクトの生成

```
minim = new Minim(this);
```

データ読み込み先の生成

```
in =  
minim.getLineIn(Minim.STEREO);
```

サウンドデータ取得

```
in.bufferSize()  
in.left.get()  
in.right.get()
```

後始末

```
in.close();  
minim.stop();  
super.stop();
```

文字列 (String)

toUpperCase



大文字に変更した文字列を返す

“KaiT2016”.toUpperCase() -> “KAIT2016”

toLowerCase



小文字に変更した文字列を返す

“KaiT2016”.toLowerCase() -> “kait2016”

授業時に配布した資料

<http://www.sato-lab.jp/imfu/index.html>

においてあります。