2017年度情報メディア基盤ユニット
7月4日講義内で作ったサンプルプログラム

```
// その1
import ddf.minim.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.signals.*;
import ddf.minim.spi.*;
import ddf.minim.ugens.*;

Minim minim;
AudioPlayer player;
void setup() {
  size(100, 100);
  minim = new Minim(this);// Miminオブジェクトの生成
  player = minim.loadFile("schoolsong.mp3");
}
void draw() {
  // draw関数がないと、keyPressed関数などが上手く動作しない
  // やりたいことを書く
}

void mouseClicked() {
  player.play();  // 1回だけ演奏
  //player.loop(); // 繰り返し演奏
  //player.play(10*1000);  // 演奏開示の時間を指定（ミリ秒）

//  player.rewind(); // 巻き戻し
//  player.play();
}

void keyPressed(){
  if(key == 'p'){
    player.pause(); // ポーズ
  }else if(key == 'r'){
    player.rewind();
  }
}
void stop() {
  player.close(); // AudioPlayerの機能を終了する
  minim.stop(); // Minimの機能を停止する
  super.stop(); // 停止の際のおまじない
}

// その2
import ddf.minim.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.signals.*;
import ddf.minim.spi.*;
import ddf.minim.ugens.*;

Minim minim;
```

```
AudioSnippet player;

void setup() {
  size(100, 100);
  minim = new Minim(this);// Mimin オブジェクトの生成
  player = minim.loadSnippet("schoolsong.mp3");
}
void draw() {
  // やりたいことを書く
  // Do nothing
}

void mouseClicked() {
  player.play();  //
  //player.loop();
  //player.play(10*1000);  //

//  player.rewind();
//  player.play();
}

void keyPressed(){
  if(key == 'p'){
    player.pause();
  }else if(key == 'r'){
    player.rewind();
  }
}
void stop() {
  player.close(); // AudioSnippet の機能を終了する
  minim.stop(); // Minim の機能を停止する
  super.stop(); // 停止の際のおまじない
}

//その 3
import ddf.minim.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.signals.*;
import ddf.minim.spi.*;
import ddf.minim.ugens.*;

Minim minim;
AudioSample player;

void setup() {
  size(100, 100);
  minim = new Minim(this);
  // 読み込むファイルが変わっています。
  player = minim.loadSample("score.wav");
}
void draw() {
  // Write what you do
}
```

```
void mouseClicked() {
  player.trigger();
}

// その4
import ddf.minim.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.signals.*;
import ddf.minim.spi.*;
import ddf.minim.ugens.*;

Minim minim;
AudioPlayer player;
void setup() {
  size(400, 100);
  minim = new Minim(this);// Miminオブジェクトの生成
  player = minim.loadFile("schoolsong.mp3");
}
void draw() {
  background(255);
  if(player.isPlaying()){
  // 曲の演奏場所や長さを取得
    float w = map(player.position(),0,player.length(),0,width);
    fill(128);
    rect(0,0,w,height);
  }
}

void mouseClicked() {
  player.play();  //
  //player.loop();
  //player.play(10*1000);  //

//  player.rewind();
//  player.play();
}

void keyPressed(){
  if(key == 'p'){
    player.pause();
  }else if(key == 'r'){
    player.rewind();
  }
}
void stop() {
  player.close(); // AudioSnippetの機能を終了する
  minim.stop(); // Minimの機能を停止する
  super.stop(); // 停止の際のおまじない
}

//その5
import ddf.minim.*;
import ddf.minim.analysis.*;
```

```
import ddf.minim.effects.*;
import ddf.minim.signals.*;
import ddf.minim.spi.*;
import ddf.minim.ugens.*;

Minim minim;
AudioSample player1,player2;

void setup() {
  size(100, 100);
  minim = new Minim(this);
  // 読み込むファイルが変わっています。
  player1 = minim.loadSample("score.wav");
  player2 = minim.loadSample("cannon.wav");
}
void draw() {
  // Write what you do
}

void keyPressed(){
  if(key == 's'){
    player1.trigger();
  }else if(key == 'c'){
    player2.trigger();
  }
}

//その6
import ddf.minim.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.signals.*;
import ddf.minim.spi.*;
import ddf.minim.ugens.*;

Minim minim;
AudioOutput out;
//SineWave sine;

void setup() {
  minim = new Minim(this);
  out = minim.getLineOut(Minim.STEREO);
 // SineWave sine = new SineWave(440, 0.5, out.sampleRate());
 // SineWave sine = new SineWave(220, 0.5, out.sampleRate());
 // SineWave sine = new SineWave(880, 0.5, out.sampleRate());
 //SquareWave sq = new SquareWave(440,0.5,out.sampleRate());
 SawWave sw = new SawWave(440,0.5,out.sampleRate());
  //out.addSignal(sine);
  //out.addSignal(sq);
  out.addSignal(sw);
}

void draw() {
}
```

```
void stop() {
  out.close(); // ライン出力の機能を終了する
  minim.stop(); // Minim の機能を停止する
  super.stop(); // 停止の際のおまじない
}

//その 7
import ddf.minim.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.signals.*;
import ddf.minim.spi.*;
import ddf.minim.ugens.*;

Minim minim;
AudioOutput out;
//SineWave sine;

void setup() {
  minim = new Minim(this);
  out = minim.getLineOut(Minim.STEREO);
  SineWave sine = new SineWave(440, 0.8, out.sampleRate());
  out.addSignal(sine);
  sine = new SineWave(880, 0.4, out.sampleRate());
  out.addSignal(sine);
   sine = new SineWave(1320, 0.2, out.sampleRate());
  out.addSignal(sine);
  // SineWave sine = new SineWave(220, 0.5, out.sampleRate());
  // SineWave sine = new SineWave(880, 0.5, out.sampleRate());

  out.addSignal(sine);
}

void draw() {
}

void stop() {
  out.close(); // ライン出力の機能を終了する
  minim.stop(); // Minim の機能を停止する
  super.stop(); // 停止の際のおまじない
}

//その 8
import ddf.minim.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.signals.*;
import ddf.minim.spi.*;
import ddf.minim.ugens.*;

Minim minim;
AudioOutput out;
SineWave sine;
```

```
void setup() {
  size(600, 200);
  minim = new Minim(this);
  out = minim.getLineOut(Minim.STEREO);
  sine = new SineWave(440, 0.5, out.sampleRate());
  out.addSignal(sine);
}
void draw() {
  background(255);
  stroke(0);
  fill(200);
  ellipse(mouseX, mouseY, 40, 40);
}
void mouseMoved() {
  // 周波数を計算する
  float freq = map(mouseY, 0, height-1, 400, 1600);
  // パンの値を計算する
  float pan = map(mouseX, 0, width, -1, 1);
  sine.setFreq(freq);// 周波数を変更する
  sine.setPan(pan); // パン位置を変更する
}
void stop() {
  out.close(); // ライン出力の機能を終了する
  minim.stop(); // Minim の機能を停止する
  super.stop(); // 停止の際のおまじない
}

//その 9
import ddf.minim.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.signals.*;
import ddf.minim.spi.*;
import ddf.minim.ugens.*;

Minim minim;
AudioInput in;
int bufferSize = 1024;
float [] buffer = new float[bufferSize];
void setup() {
  size(400, 230);
  minim = new Minim(this);
  in = minim.getLineIn(Minim.STEREO);
}
void draw() {
  background(255);
  stroke(0);
  beginShape();
  for (int i = 0; i < in.bufferSize(); i++) {
    float x=map(i, 0, in.bufferSize(), 0, width-1);
    vertex(x, 60 + in.left.get(i)*50);
  }
  endShape();
  beginShape();
```

```
  for (int i = 0; i < in.bufferSize(); i++) {
    float x=map(i, 0, in.bufferSize(), 0, width-1);
    vertex(x, 170 + in.right.get(i)*50);
  }
  endShape();
}
void stop() {
  in.close();// AudioPlayer の機能を終了する
  minim.stop(); // Minim の機能を停止する
  super.stop(); // 停止の際のおまじない
}
```