

7月23日の授業中に作成したスケッチ

```
//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); // Minim オブジェクトの生成
  player = minim.loadFile("schoolsong.mp3");
}

void draw() {
  // やりたいことを書く
  background(255);
  line(frameCount % width, 0, frameCount % width, height);
}

void keyPressed() {
  if (key == 's') {
    player.pause();
  } else if (key == 'r') {
    player.rewind();
  } else if (key == 'p') {
    player.play();
  }
}

void mouseClicked() {
  //player.play();
  player.loop();
}

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); // Minim 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
  background(255);
  line(frameCount % width, 0, frameCount % width, height);
}

void keyPressed() {
  if (key == 's') {
    player.pause();
  } else if (key == 'r') {
    player.rewind();
  } else if (key == 'p') {
    player.play();
  }
}

void mouseClicked() {
  player.play();
}

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); // Minim オブジェクトの生成
  player = minim.loadSample("score.wav");
}

void draw() {
  // やりたいことを書く
  background(255);
}
```

```
    line(frameCount % width,0,frameCount % width,height);
    if(frameCount % 120 == 0){
        player.trigger();
    }
}

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

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(100, 100);
    minim = new Minim(this); // Minim オブジェクトの生成
    player = minim.loadFile("schoolsong.mp3");
}

void draw() {
    // やりたいことを書く
    background(255);
    if (player.isPlaying()) {
        stroke(255, 0, 0);
    } else {
        stroke(0);
    }
    line(frameCount % width, 0, frameCount % width, height);
}

void keyPressed() {
    if (key == 's') {
        player.pause();
    } else if (key == 'r') {
        player.rewind();
    } else if (key == 'p') {
        player.play();
    }
}
}
```

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

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;
AudioOutput out;
SineWave sine;

void setup() {
    size(100,100);
    minim = new Minim(this);
    out = minim.getLineOut(Minim.STEREO);
    sine = new SineWave(880, 0.9, 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();
}

//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 sine1,sine2,sine3;
```

```
void setup() {
  size(100,100);
  minim = new Minim(this);
  out = minim.getLineOut(Minim.STEREO);
  sine1 = new SineWave(440, 0.5, out.sampleRate());
  sine2 = new SineWave(880, 0.2, out.sampleRate());
  sine3 = new SineWave(1760, 0.1, out.sampleRate());
  out.addSignal(sine1);
  out.addSignal(sine2);
  out.addSignal(sine3);
}

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();
}

//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;
SquareWave square;

void setup() {
  size(100,100);
  minim = new Minim(this);
  out = minim.getLineOut(Minim.STEREO);
  square = new SquareWave(440, 0.5, out.sampleRate());
  out.addSignal(square);
}

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();
}

//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;
SawWave saw;

void setup() {
  size(100,100);
  minim = new Minim(this);
  out = minim.getLineOut(Minim.STEREO);
  saw = new SawWave(440, 0.5, out.sampleRate());
  out.addSignal(saw);
}

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();
}

//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;
AudioSample[] se;

void setup() {
  size(100, 100);
  minim = new Minim(this); // Minim オブジェクトの生成
  se = new AudioSample[4];
  se[0] = minim.loadSample("appear01.wav");
  se[1] = minim.loadSample("appear02.wav");
  se[2] = minim.loadSample("appear03.wav");
  se[3] = minim.loadSample("appear04.wav");
}

void draw() {
  // やりたいことを書く
}

void keyPressed() {
  if (key=='1') {
```

```
        se[0].trigger();
    } else if (key=='2') {
        se[1].trigger();
    } else if (key=='3') {
        se[2].trigger();
    } else if (key=='4') {
        se[3].trigger();
    }
}

void stop() {
    // AudioSample の機能を終了する
    for (int i=0; i<se.length; i++) {
        se[i].close();
    }
    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10
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  
}
```