Design Programming

Spring 2018 - 10

2018.6.20 Keio University, SFC

動的配列を扱う

```
ArrayList <Integer> x, y;

void setup() {
    size(500, 500);
    x = new ArrayList<Integer>();
    y = new ArrayList<Integer>();
}

void draw() {
    for (int i = 0; i < x.size(); i++) {
        ellipse(x.get(i), y.get(i), 10, 10);
    }
}

void mouseReleased() {
    x.add(mouseX);
    y.add(mouseY);
}
```

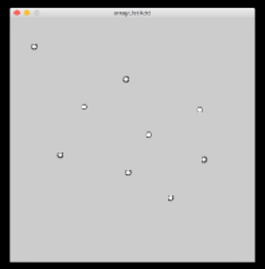
動的配列を宣言する

インスタンスを生成する

要素を追加する

要素数でforループを回す

.get(インデックス)で要素を指定する



10/arrayListAdd/arrayListAdd.pde

演習 1

さきほどのスケッチを利用して，キー「c」を押して離したら
すべてのパーティクルの要素が消される（クリアされる）
機能を実装しなさい.

ヒント

`.clear()` を使うとすべての要素がクリアされる

演習 1

```
void keyReleased() {
    if (key == 'c') {
            x.clear();
            y.clear();
    }
}
```

10/practice1/practice1.pde

演習 2

さきほどのスケッチを利用して，キー「r」を押して離したら

先頭の要素が１つ消される機能を実装しなさい.

ヒント

`.remove(i)`　を使うとi番目（0から）の要素がクリアされる

演習 2

```
void keyReleased() {
    if (key == 'r') {
          x.remove(0);
          y.remove(0);
      }
}
```

10/practice2/practice2.pde

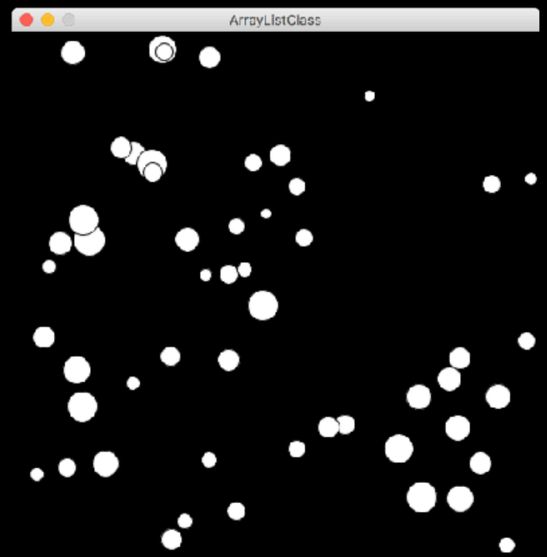
自作クラスにも適用する

```
ArrayList<Particle> p;

void setup() {
    size(500, 500);
    p = new ArrayList<Particle>();
}

void draw() {
    background(0);
    for (int i = 0; i < p.size(); i++) {
        p.get(i).update();
        p.get(i).display();
    }
}

void mouseReleased() {
    p.add(new Particle(mouseX, mouseY,
    random(10, 30), random(1, 5), random(1, 5)));
}
```



クリックするたびに動くパーティクルが増えていく

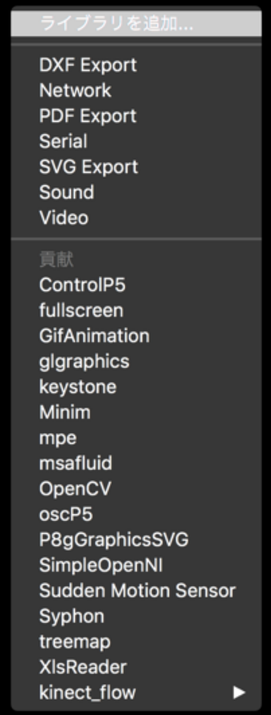
10/arrayListClass/arrayListClass.pde

サウンドメディアを扱うには

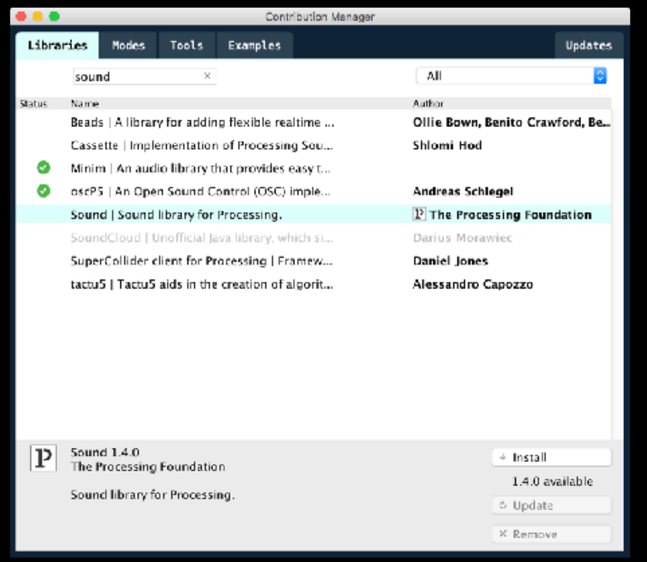
ライブラリ（標準では不可能な拡張機能を使用するためのプログラム郡）を使用する.

スケッチ＞

ライブラリをインポート



ライブラリを追加



Librariesタブ＞Sound

をInstall

ライブラリについて

かなりの数のライブラリが存在するため,

たいていのことがProcessingを使用すれば可能になる.

https://processing.org/reference/libraries/index.html

センサーの値を読み込みたい

pdfに書き出したい

３Dモデルデータを読み込みたい

gifに書き出したい

他のソフトウェアから操作したい

外部デバイスを読み込みたい

サウンドメディアを扱う

```
import processing.sound.*;
SoundFile sf;

void setup() {
    sf = new SoundFile(this, "sample.mp3");
}

void draw() {
}

void  keyPressed() {
    if (key == ENTER) sf.play();
}
```

サウンドライブラリをインポートする．「＊」は任意のファイルを示す

サウンドファイル用のオブジェクトを宣言

ファイル名を指定して，インスタンスを生成する．

thisによって，SoundFileオブジェクトのメンバ変数を参照．

再生用の関数play()で再生

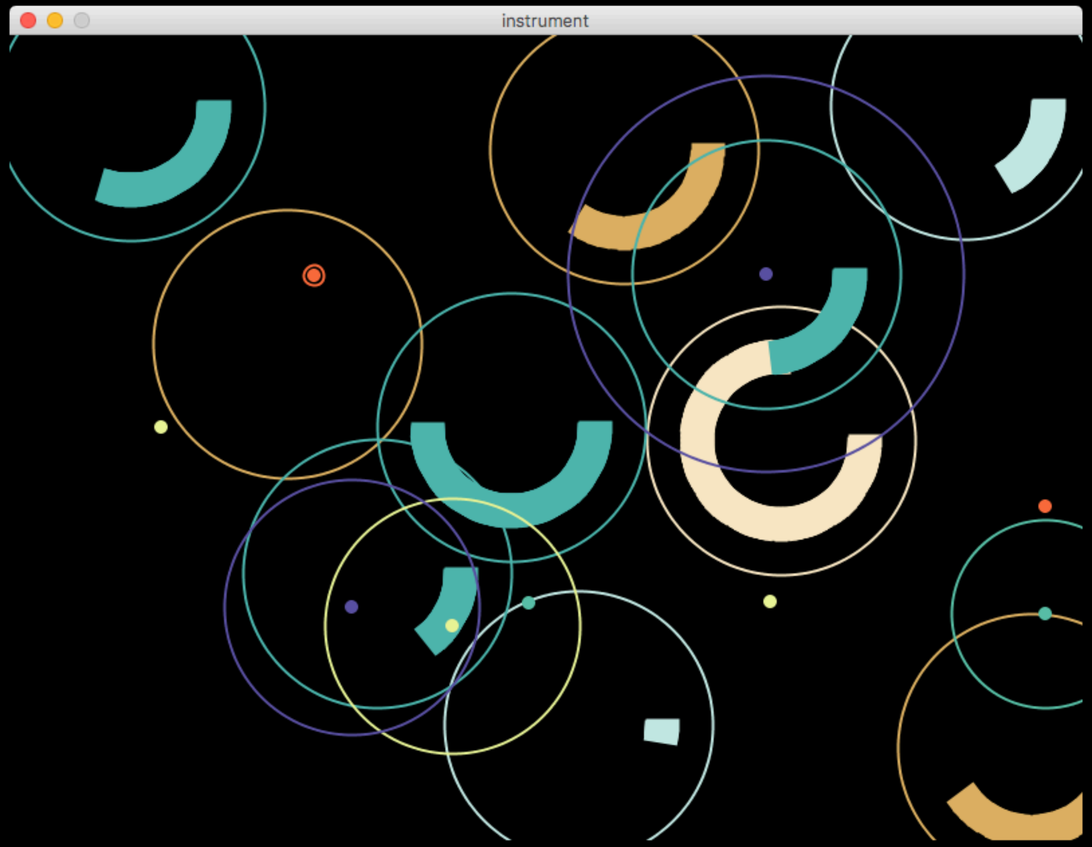
10/soundPlayer/soundPlayer.pde

サウンドファイルクラスの

メンバ関数

関数名	関数の意味
`loop();`	ループ再生
`jump();`	指定した秒から再生
`play();`	はじめから再生
`rate();`	再生速度を変更
`amp();`	音量を変更

楽器（のようなもの）をつくる



プレイヤーの数を変える

```
import processing.sound.*;
SoundFile bd;

ArrayList<SoundPlayer> sp;

void setup() {
    size(800, 600);
    sp = new ArrayList<SoundPlayer>();
    bd = new SoundFile(this, "bd1.aif");
}

void draw() {
    background(0);
    for (int i = 0; i < sp.size(); i++) {
        if (sp.get(i).count == 0)
            bd.play();
        sp.get(i).draw();
    }
}

void keyReleased() {
    sp.add(new SoundPlayer(mouseX, mouseY));
}
```

10/soundAdd/soundAdd.pde

```
class SoundPlayer {
    SoundPlayer(int _x, int _y) {
        x = _x;
        y = _y;
        d = 200;
        count = 0;
    }
    int x;
    int y;
    int d;
    int count;
    void draw() {
        count++;
        ellipse(x, y, d, d);
    }
}
```

10/soundAdd/soundPlayer.pde

周期をつくる

```
class SoundPlayer {
    SoundPlayer(int _x, int _y) {
        x = _x;
        y = _y;
        d = 200;
        count = 0;
        c = color(255);
    }
    int x;
    int y;
    int d;
    int count;
    color c;
```

```
    void draw() {
            strokeCap(SQUARE);
            noFill();
            stroke(c);
            strokeWeight(2);
            count++;
            ellipse(x, y, d, d);
            if (count == 60)
                count = 0;
            strokeWeight(10);
            for (int i = 0; i < count; i++) {
                float co = cos(radians(float(i) * 6));
                float si = sin(radians(float(i) * 6));
                line(x + 50 * co, y + 50 * si, x + 75 * co, y + 75 * si);
            }
        }
    }
```

10/soundCycle/soundPlayer.pde

入力で周期を変化させる

```
class SoundPlayer {
    SoundPlayer(char _pressedKey, int _x, int _y) {
        pressedKey = _pressedKey;
        x = _x;
        y = _y;
        d = 200;
        count = 0;
        switch (pressedKey) {
        case '1':
            mode = 1;
            c = color(216, 179, 101);
            break;
        case '2':
            mode = 2;
            c = color(246, 232, 195);
            break;
        case '3':
            mode = 3;
            c = color(199, 234, 229);
            break;
        case '4':
            mode = 4;
            c = color(90, 180, 172);
            break;
        }
    }
    int x;
    int y;
    int d;
    int count;
    color c;
    int mode;
    char pressedKey;
    void draw() {
        strokeCap(SQUARE);
        noFill();
        stroke(c);
        strokeWeight(2);
        count++;
        ellipse(x, y, d, d);
        if (count == 60 * mode)
            count = 0;
        strokeWeight(10);
        for (int i = 0; i < count; i++) {
            float co = cos(radians(float(i) * 6
            / float(mode)));
            float si = sin(radians(float(i) * 6
            / float(mode)));
            line(x + 50 * co, y + 50 * si,
                 x + 75 * co, y + 75 * si);
        }
    }
}
```

10/soundCycleChange/soundPlayer.pde

入力で周期を変化させる

```
import processing.sound.*;
SoundFile bd;

ArrayList<SoundPlayer> sp;

void setup() {
    size(800, 600);
    sp = new ArrayList<SoundPlayer>();
    bd = new SoundFile(this, "bd1.aif");
}

void draw() {
    background(0);
    for (int i = 0; i < sp.size(); i++) {
        if (sp.get(i).count == 0)
            bd.play();
        sp.get(i).draw();
    }
}

void keyReleased() {
    if (key == '1' || key == '2' || key == '3' || key == '4')
        sp.add(new SoundPlayer(key, mouseX, mouseY));
}
```

10/soundCycleChange/soundCycleChange.pde

物体がプレイヤーエリア内で鳴る

```
class SoundObject {
    SoundObject(int _x, int _y) {
        x = _x;
        y = _y;
        vx = (int)random(-2, 2);
        vy = (int)random(-2, 2);
        d = 10;
        c = color(255);
    }
    int x;
    int y;
    int d;
    int vx;
    int vy;
    color c;
```

```
    void update() {
            x += vx;
            y += vy;
            if (x >= width - d / 2 || x <= d / 2)
                vx = -vx;
            if (y >= height - d / 2 || y <= d / 2)
                vy = -vy;
        }
        void draw() {
            noStroke();
            fill(c);
            ellipse(x, y, d, d);
        }
}
```

10/soundArea/soundObject.pde

物体がプレイヤーエリア内で鳴る

```
import processing.sound.*;
SoundFile bd;

ArrayList<SoundPlayer> sp;
ArrayList<SoundObject> so;

void setup() {
    size(800, 600);
    sp = new ArrayList<SoundPlayer>();
    so = new ArrayList<SoundObject>();
    bd = new SoundFile(this, "bd1.aif");
}
```

```
void draw() {
    background(0);
    for (int i = 0; i < sp.size(); i++) {
        if (sp.get(i).count == 0)
            for (int j = 0; j < so.size(); j++) {
                if (dist(sp.get(i).x, sp.get(i).y,
                    so.get(j).x, so.get(j).y)
                    < sp.get(i).d / 2)
                    bd.play();
            }
        sp.get(i).draw();
    }
    for (int i = 0; i < so.size(); i++) {
        so.get(i).update();
        so.get(i).draw();
    }
}

void keyReleased() {
    if (key == '1' || key == '2' || key == '3' || key == '4')
        sp.add(new SoundPlayer(key, mouseX, mouseY));
    if (key == '5')
        so.add(new SoundObject(mouseX, mouseY));
}
```

10/soundArea/soundArea.pde

音色を増やす

```
class SoundObject {
    SoundObject(char _pressedKey, int _x, int _y) {
        x = _x;
        y = _y;
        vx = (int)random(-2, 2);
        vy = (int)random(-2, 2);
        d = 10;
        pressedKey = _pressedKey;
        switch (pressedKey) {
        case '5':
            c = color(94, 79, 162);
            break;
        case '6':
            c = color(102, 194, 165);
            break;
        case '7':
            c = color(230, 245, 152);
            break;
        case '8':
            c = color(244, 109, 67);
            break;
        case '9':
            c = color(50, 136, 189);
            break;
        }
    }
```

```
int x;
    int y;
    int d;
    int vx;
    int vy;
    color c;
    char pressedKey;
    void update() {
        x += vx;
        y += vy;
        if (x >= width - d / 2 || x <= d / 2)
            vx = -vx;
        if (y >= height - d / 2 || y <= d / 2)
            vy = -vy;
    }
    void draw() {
        noStroke();
        fill(c);
        ellipse(x, y, d, d);
    }
}
```

10/soundColor/soundObject.pde

音色を増やす

```
import processing.sound.*;
SoundFile bd, sn, hh, st1, st2;

ArrayList<SoundPlayer> sp;
ArrayList<SoundObject> so;

void setup() {
    size(800, 600);
    sp = new ArrayList<SoundPlayer>();
    so = new ArrayList<SoundObject>();
    bd = new SoundFile(this, "bd1.aif");
    sn = new SoundFile(this, "sn1.aif");
    hh = new SoundFile(this, "hh1.aif");
    st1 = new SoundFile(this, "st1.wav");
    st2 = new SoundFile(this, "st2.wav");
}
```

10/soundColor/soundColor.pde

```
void draw() {
    background(0);
    for (int i = 0; i < sp.size(); i++) {
        if (sp.get(i).count == 0)
            for (int j = 0; j < so.size(); j++) {
                if (dist(sp.get(i).x, sp.get(i).y, so.get(j).x, so.get(j).y) < sp.get(i).d / 2)
                    switch (so.get(j).pressedKey) {
                    case '5':
                        bd.play();
                        break;
                    case '6':
                        sn.play();
                        break;
                    case '7':
                        hh.play();
                        break;
                    case '8':
                        st1.play();
                        break;
                    case '9':
                        st2.play();
                        break;
                    }
            }
        sp.get(i).draw();
    }
    for (int i = 0; i < so.size(); i++) {
        so.get(i).update();
        so.get(i).draw();
    }
}
```

10/soundColor/soundColor.pde

音色を増やす

```
void keyReleased() {
    if (key == '1' || key == '2' || key == '3' || key == '4')
        sp.add(new SoundPlayer(key, mouseX, mouseY));
    if (key == '5' || key == '6' || key == '7' || key == '8' || key == '9')
        so.add(new SoundObject(key, mouseX, mouseY));
}
```

10/soundColor/soundColor.pde

音が鳴るときにアニメーション

```
class SoundObject {
    SoundObject(char _pressedKey, int _x, int _y) {
        x = _x;
        y = _y;
        vx = (int)random(-2, 2);
        vy = (int)random(-2, 2);
        d = 10;
        pressedKey = _pressedKey;
        switch (pressedKey) {
        case '5':
            c = color(94, 79, 162);
            break;
        case '6':
            c = color(102, 194, 165);
            break;
        case '7':
            c = color(230, 245, 152);
            break;
        case '8':
            c = color(244, 109, 67);
            break;
        case '9':
            c = color(50, 136, 189);
            break;
        }
        count = 0;
        wave = false;
    }
    int x;
    int y;
    int d;
    int vx;
    int vy;
    color c;
    char pressedKey;
    Boolean wave;
    int count;
--update関数省略
    void draw() {
        noStroke();
        fill(c);
        ellipse(x, y, d, d);
        if (wave) {
            count+=5;
            noFill();
            strokeWeight(2);
            stroke(c);
            ellipse(x, y, d + count, d + count);
            if (count >= 60 * 5) {
                count = 0;
                wave = false;
            }
        }
    }
}
```

10/instrument/SoundObject.pde

音が鳴るときにアニメーション

```
void draw() {
    background(0);
    for (int i = 0; i < sp.size(); i++) {
        if (sp.get(i).count == 0)
            for (int j = 0; j < so.size(); j++) {
                if (dist(sp.get(i).x, sp.get(i).y, so.get(j).x, so.get(j).y) < sp.get(i).d / 2) {
                    switch (so.get(j).pressedKey) {
                    case '5':
                        bd.play();
                        break;
                    case '6':
                        sn.play();
                        break;
                    case '7':
                        hh.play();
                        break;
                    case '8':
                        st1.play();
                        break;
                    case '9':
                        st2.play();
                        break;
                    }
                    so.get(j).wave = true;
                }
```

以下省略

10/instrument/instrument.pde

課題

- 今日学習した「動的配列を扱う，メディアを扱う」を使用してスケッチを１つアップロードしなさい.
  アップロード先は「10 arrayList media」とする.
  ※今日学習していない分野でもすでに自分が知っている技術は使用可能とする.

  〆切：6月26日24時まで

次回

- 応用表現 2