

VJのTips

キーボードによる画像の切替

[vj_sample01] キーボードによる画像表示1

任意のキーを押したときに画像を表示させる

```
boolean button1;
//キーボードを押すことによって
//trueとfalseが入れ替わる変数
float b1;//図形の角度を変更する変数

void setup(){
  size(400,400);
  smooth();
  frameRate(30);
}

void draw(){
  background(255);

  if (button1){//ボタンが押された後の処理
    pushMatrix();
    b1 = b1+3;
    if(360<b1)b1=0;
    translate(width/2,height/2);
    rotate(radians(b1));
    noFill();
    strokeWeight(10);
    stroke(0,50);
    rect(-100,-100,200,200);
    rect(-200,-200,400,400);
    popMatrix();
  }
}

void keyPressed(){ //キーボードを押した瞬間の処理
  button1 = true;
}
void keyReleased(){//キーボードを離した瞬間の処理
  button1 = false;
}
```

[vj_sample02] キーボードによる画像表示2

1と2と3のキーを押した時に画像を表示させる

```
boolean button1, button2, button3;
float b1,b2,b3;

void setup(){
  size(400,400);
  smooth();
  frameRate(30);
}

void draw(){
  noStroke();
  fill(255,100);
  rect(0,0,width,height);

  if (button1){//1のキーが押されたときの処理
    drawButton1();
  }
  else{
    b1=0;
  }
  if (button2){//2のキーが押されたときの処理
    drawButton2();
  }
  if (button3){ //2のキーが押されたときの処理
    drawButton3();
  }
}
```

(vj_sample02続き)

```
void drawButton1(){//1のキーが押されたときの処理内容
  noFill();
  strokeWeight(10);
  stroke(0,50);
  ellipse(width/2,height/2,b1,b1);
  b1=b1+10;
  if(b1 > height)b1=0;
}

void drawButton2(){//2のキーが押されたときの処理内容
  pushMatrix();
  b2 = b2+3;
  if(360<b2)b2=0;
  translate(width/2,height/2);
  rotate(radians(b2));
  noFill();
  strokeWeight(10);
  stroke(0,50);
  rect(-100,-100,200,200);
  rect(-200,-200,400,400);
  popMatrix();
}

void drawButton3(){//3のキーが押されたときの処理内容
  strokeWeight(10);
  noFill();
  stroke(0,50);
  line(b3,0,b3,height);
  b3=b3+5;
  if(b3==width)b3=0;
}

void keyPressed(){//キーボードを押した瞬間の処理
  switch(key){
    case('1')://キーが1の場合
      button1 = true;
      break;
    case('2')://キーが2の場合
      button2 = true;
      break;
    case('3')://キーが3の場合
      button3 = true;
      break;
  }
}

void keyReleased(){//キーボードを離した瞬間の処理
  switch(key){
    case('1')://キーが1の場合
      button1 = false;
      break;
    case('2')://キーが2の場合
      button2 = false;
      break;
    case('3')://キーが3の場合
      button3 = false;
      break;
  }
}
```

[vj_sample03] キーを押したとき順次画像が切り替わる

キー1を押すごとに順番に切り替わり、キー2は逆方向に切り替わる。

```
boolean button1, button2, button3;
float b1,b2,b3;
int keyOrder=0;

void setup(){
  size(400,400);
  smooth();
  frameRate(30);
}

void draw(){
  noStroke();
  //残像を残す処理
  fill(255,160);
  rectMode(CORNER);
  rect(0,0,width,height);

  //keyOrder変数の値に応じて処理を変更する
  switch(keyOrder){
    case(0):
      drawButton1();
      break;
    case(1):
      drawButton2();
      break;
    case(2):
      drawButton3();
      break;
    case(3):
      drawButton1();
      drawButton2();
      drawButton3();
      break;
  }
}
```

(vj_sample03続き)

```
void drawButton1(){//1のキーが押されたときの処理内容
  noFill();
  strokeWeight(3);
  stroke(0,80);
  ellipse(width/2,height/2,b1,b1);
  b1=b1+10;
  if(b1 > height)b1=0;
}

void drawButton2(){//2のキーが押されたときの処理内容
  pushMatrix();
  b2 = b2+3;
  if(360<b2)b2=0;
  translate(width/2,height/2);
  rotate(radians(b2));
  noFill();
  strokeWeight(3);
  stroke(0,80);
  rect(-100,-100,200,200);
  rect(-200,-200,400,400);
  popMatrix();
}

void drawButton3(){//3のキーが押されたときの処理内容
  strokeWeight(3);
  noFill();
  stroke(0,80);
  line(b3,0,b3,height);
  b3=b3+5;
  if(b3==width)b3=0;
}

//キーが押されたときに変数keyOrderをカウントアップ
void keyPressed(){
  if(key=='1'){
    keyOrder++;
    if(keyOrder > 3) keyOrder=0;
  }
}

//キーが離されたときに変数keyOrderをカウントダウン
void keyReleased(){
  if(key=='2'){
    keyOrder--;
    if(keyOrder < 0) keyOrder=3;
  }
}
```

```

//クラスの定義
int numBall = 200;
Ball[] myBall = new Ball[numBall];
Spot sp1, sp2;

boolean button1, button2, button3;
float b1,b2,b3;

void setup(){
  size(400,400);
  smooth();
  colorMode(HSB);
  frameRate(30);

//クラスの初期化
for(int i =0; i<myBall.length ; i++){
  myBall[i] = new Ball(color(random
    (255),255,200),0,i*2,random(20));
}
sp1 = new Spot(100, 50, 100, 0.5);
sp2 = new Spot(250, 100, 200, 2.0);
}

void draw(){
  background(255);

if (button1){ //キー1によるクラスの呼び出し
  sp1.move();
  sp2.move();
  sp1.display();
  sp2.display();
}

if (button2){ //キー2によるクラスの呼び出し
  for( int i = 0; i<myBall.length; i++){
    myBall[i].move();
    myBall[i].display();
  }
}

void keyPressed(){ //キーボードを押した瞬間の処理
  switch(key){
    case('1'):
      button1 = true;
      break;
    case('2'):
      button2 = true;
      break;
  }
}

void keyReleased(){ //キーボードを離した瞬間の処理
  switch(key){
    case('1'):
      button1 = false;
      break;
    case('2'):
      button2 = false;
      break;
  }
}

```

```

class Ball {
  color c;
  float xpos;
  float ypos;
  float xspeed;

  Ball(color tempC, float tempXpos,
    float tempYpos, float tempXspeed) {
    c = tempC;
    xpos = tempXpos;
    ypos = tempYpos;
    xspeed = tempXspeed;
  }

  void display() {
    noStroke();
    fill(c,80);
    ellipse(xpos,ypos,20,20);
  }

  void move() {
    xpos = xpos + xspeed;
    if (xpos > width+100) {
      xpos = -100;
    }
  }
}

```

```

class Spot {
  float x, y;
  float diameter;
  float speed;
  int direction = 1;

  Spot(float xpos, float ypos, float dia, float sp) {
    x = xpos;
    y = ypos;
    diameter = dia;
    speed = sp;
  }

  void move() {
    y += (speed * direction);
    if ((y > (height - diameter / 2)) || (y < diameter / 2)) {
      direction *= -1;
    }
  }

  void display() {
    noFill();
    strokeWeight(2);
    stroke(90);
    ellipse(x, y, diameter, diameter);
  }
}

```

[vj_sample06] 特定のクラスの図形のみ残像を残す
キー1で残像のある図形を表示

```
//クラスの定義
Sample mySample1, mySample2;
Spot sp1;

void setup(){
  size(400,400);
  frameRate(30);
//クラスの初期化
  mySample1 = new Sample
    (color(255,0,0,100),0,120,2);
  mySample2 = new Sample
    (color(255,220,0,100),0,250,8);
  sp1 = new Spot(250, 100, 200, 6.0);
  smooth();
}

void draw(){

  background(255);

//クラスの描写
  sp1.move();
  sp1.display();

  mySample1.display();
  mySample1.move();
  mySample2.display();
  mySample2.move();
}

void keyPressed(){//キーボードを押した瞬間の処理
  if(key=='1'){
    //クラス内のfadein関数を呼出
    mySample1.fadeIn();
    mySample2.fadeIn();
  }
}

void keyReleased(){//キーボードを離した瞬間の処理
  if(key=='1'){
    //クラス内のfadeOut関数を呼出
    mySample1.fadeOut();
    mySample2.fadeOut();
  }
}
```

(vj_sample06続き)

```
class Sample {
  color c;
  float xpos;
  float ypos;
  float xspeed;
  int f;
  boolean fade;

  Sample(color tempC, float tempXpos,
         float tempYpos, float tempXspeed) {
    c = tempC;
    xpos = tempXpos;
    ypos = tempYpos;
    xspeed = tempXspeed;
    f = 0;
    fade = false;
  }

  void display() {

    if(fade==true){
      f=f+10;
    }else{
      f=f-10;
    }
    if(f<0)f=0;
    if(f>255)f=255;

    noStroke();
    fill(c,f);
    rectMode(CENTER);
    rect(xpos,ypos,200,200);
  }

  void move() {
    xpos = xpos + xspeed;
    if (xpos > width+200) {
      xpos = -200;
    }
  }
  void fadeIn(){
    fade = true;
  }
  void fadeOut(){
    fade = false;
  }
}
```

```
class Spot {
  float x, y;
  float diameter;
  float speed;
  int direction = 1;

  Spot(float xpos, float ypos, float dia, float sp) {
    x = xpos;
    y = ypos;
    diameter = dia;
    speed = sp;
  }

  void move() {
    y += (speed * direction);
    if ((y > (height - diameter / 2)) || (y < diameter / 2)) {
      direction *= -1;
    }
  }

  void display() {
    noFill();
    strokeWeight(2);
    stroke(90);
    ellipse(x, y, diameter, diameter);
  }
}
```