

# 動画の作成2

担当: 佐藤

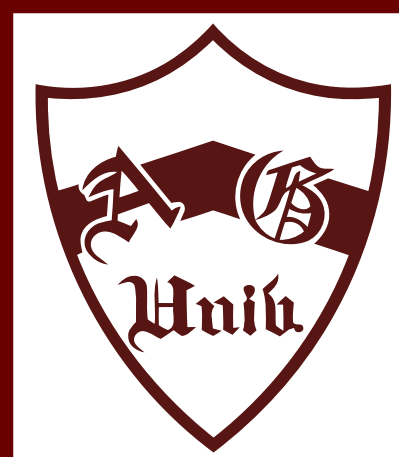
## 計算機実習III

第3回: 演習課題解説



[sato@ise.aoyama.ac.jp](mailto:sato@ise.aoyama.ac.jp)

2019/5/7



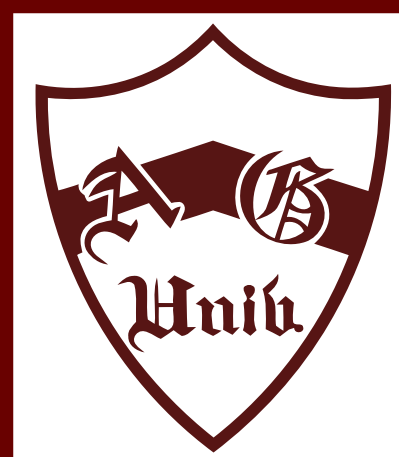
# 課題3-1

## 解答例

```
float rectSize = 50 ;  
void setup() {  
  size(500, 500) ;  
  background(255) ;  
  colorMode(HSB) ;  
  noStroke() ;  
}  
void draw() {  
  float rectX = int(mouseX / rectSize) * rectSize ;  
  float rectY = int(mouseY / rectSize) * rectSize ;  
  fill(255, 40) ;  
  rect(0, 0, width, height) ;  
  fill(frameCount % 256, 255, 255) ;  
  rect(rectX, rectY, rectSize, rectSize) ;  
}
```

何番目の区画か

0~255



## 課題3-2

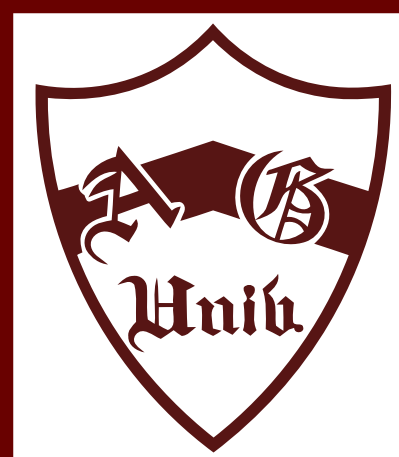
### 解答例

```
void setup() {  
  size(500, 500) ;  
  noStroke() ;  
  fill(0) ;  
}  
  
void draw() {  
  background(255) ;  
  for (int i = 0 ; i < 10 ; i++) {  
    int diameter = 5 * (i + 1) ;  
    float theta = radians(frameCount % 360 + i * 30) ;  
    ellipse(100 * cos(theta) + mouseX,  
           100 * sin(theta) + mouseY,  
           diameter, diameter) ;  
  }  
}
```

円の直径: 5~50

中心角: 30°

マウスカーソル中心



# 課題3-3

## 解答例

```
float centerX, centerY ;
float sign = 1 ;
float theta = 0 ;
float diameter = 100 ;

void setup() {
  size(150, 150) ;
  noStroke() ;
  fill(255, 255, 0) ;
}

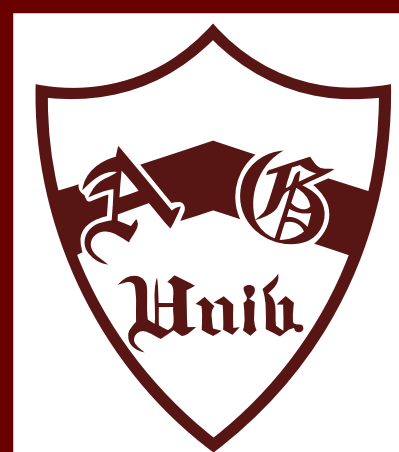
void draw() {
  background(0) ;
  setMouthAngle() ;
  drawPackman() ;
}
```

```
void setMouthAngle() {
  theta += sign * radians(1) ; // 1[rad / frame]
  if (theta > PI / 6) {
    sign = -sign ; // sign *= -1;
    theta = PI / 6 ;
  } else if (theta < 0) {
    sign = -sign ; // sign *= -1;
    theta = 0 ;
  }
}
```

□の開き角の変化: 2[rad/frame],  
□の開き角の範囲: 0~60[deg]より,  
thetaを1[rad/frame]で0⇔30[deg]  
変化させる

上下にtheta□を開く

```
void drawPackman() {
  arc(width/2, height/2, diameter, diameter, theta, TWO_PI - theta) ;
}
```



# 課題3-4

## 解答例

```
int n = 5 ;
float unitAngle = TWO_PI / n ;
float[] vertexX = new float[n] ;
float[] vertexY = new float[n] ;
int index = 0 ;
float diameter = 30 ;
int count = 0 ;
int maxCount = 300 / n ;
```

```
void setup() {
  size(500, 500) ;
  background(255) ;
  noStroke() ;
  for (int i = 0 ; i < n ; i++) {
    vertexX[i] = width / 2 + 200 * cos(i * unitAngle) ;
    vertexY[i] = height / 2 + 200 * sin(i * unitAngle) ;
  }
  colorMode(HSB) ;
}
```

頂点間の移動にかかるフレーム数

頂点間の移動におけるフレーム数の割合に対応するx座標の割合

```
void draw() {
  float x = map(count, 0, maxCount - 1,
    vertexX[index], vertexX[(index + 3) % n]) ;
  float y = map(count, 0, maxCount - 1,
    vertexY[index], vertexY[(index + 3) % n]) ;
  fill(map(index, 0, n - 1, 0, 200), 255, 255) ;
  ellipse(x, y, diameter, diameter) ;
  fill(0) ;
  ellipse(x, y, diameter / 2, diameter / 2) ;
  count = (count + 1) % maxCount ;
  if (count == 0) index = (index + 3) % n ;
  if (frameCount == 300) noLoop() ;
}
```

移動先の頂点