

Processing – Referans Kağıdı

Barış Aktemur, Özyeğin Üniversitesi

Koordinat Sistemi

- Pencerenin sol üst köşesi koordinat sisteminin orijini.
- Yatay koordinat (x koordinatı) sağa doğru büyür.
- Dikey koordinat (y koordinatı) aşağı doğru büyür.
- `size(w,h)`: Açılacak pencerenin genişliğini `w` piksel, yüksekliğini `h` piksel olarak tanımlar.

Şekiller

- `line(x1,y1,x2,y2)`: Başlangıç noktası `x1,y1`, bitiş noktası `x2,y2` olan bir çizgi çizer.
- `rect(x,y,w,h)`: Sol üst köşesi `x,y` noktasında olan, genişliği `w` piksel, yüksekliği `h` piksel olan bir dikdörtgen çizer.
- `ellipse(x,y,w,h)`: Merkezi `x,y` noktasında olan, genişliği `w` piksel, yüksekliği `h` piksel olan bir elips çizer.
- Yazdığımız komutlar yukarıdan aşağıya doğru, sırayla çalıştırılır.

Renkler

- Renkler kırmızı, mavi ve yeşil bileşenlerin karışımı olarak tanımlanır. Buna RGB uzayı denir.
- Her bir bileşen değeri `[0,255]` aralığındadır.
- `fill(r,g,b)`: Çizilecek dikdörtgen ve elipslerin iç boyasını `r,g,b` olarak tanımlar.
- `stroke(r,g,b)`: Çizilecek çizgilerin ve dikdörtgen ile elipslerin çerçevelerinin boyasını `r,g,b` olarak tanımlar.
- `background(r,g,b)`: Ekranda o ana kadar çizilmiş ne varsa siler, arkaplan rengini `r,g,b` yapar.
- `noFill()`: Çizilecek dikdörtgen ve elipslerin içlerinin boş olmasını sağlar.
- `noStroke()`: Çizilecek dikdörtgen ve elipslerin çerçevesiz olmasını sağlar.
- `strokeWeight(w)`: Çizilecek çizgilerin ve dikdörtgen ile elipslerin çerçevelerinin kalınlığının `w` piksel olmasını sağlar.
- `smooth()`: Çizgilerin daha yumuşak görünmesini sağlar.

Değişkenler

- Değişken tanımlama örneği:

```
int genislik = 80;
```

- Değişkene yeni değer atama örneği:

```
genislik = 180;
```

- Değişkenin değerini 10 artırma örneği:

```
genislik = genislik + 10;
```

- Processing'de hazır tanımlanmış iki değişken:
 - `width`: Pencerenin genişliği
 - `height`: Pencerenin yüksekliği

Setup ve draw metodları

- `setup()` metodu program başlangıcında ve sadece bir kere çalıştırılır. Ekran boyutunu tanımlayan `size()` gibi tek seferlik komutlar buraya yazılır.
- `draw()` metodu tekrar tekrar çalıştırılır.
- `mouseX` ve `mouseY` değişkenleri fare imlecinin o an pencere üzerindeki konumunu verir.
- Şekillerimizin yerini kontrol eden değişkenleri `setup()` ve `draw()`'un içinde değil dışında tanımlamalıyız.

İmajlar

- Kullanacağınız resim dosyasını (jpg, png ya da gif), programınızı kaydettiğiniz klasörün içine koyun.
- `setup()` ve `draw()` dışında bir `PImage` değişkeni tanımlayın; bu değişkene `setup()` içinde `loadImage()` kullanarak resminizi yükleyin; `draw()` içinde `image()` fonksiyonunu kullanarak resmi istediğiniz konuma çizdirin. Temel kod yapısı:

```
void setup() {
  size(500, 300);
  istanbul = loadImage("istanbul.jpg");
}

PImage istanbul;

void draw() {
  image(istanbul, 10, 40);
}
```

Rastgele sayılar

- `random(n)` bize `[0, n)` aralığında rastgele bir ondalıklı sayı verir.
- `random(m, n)` bize `[m, n)` aralığında rastgele bir ondalıklı sayı verir.
- Ondalıklı sayıyı tam sayıya `int()` ile çeviririz.

Seçim yapmak

Komutları belli bir koşul doğru olduğunda çalıştırmak, yanlış olduğunda atlamak için:

```
if (kosul) {
  komut1;
  komut2;
  ...
}
```

Belli bir koşul doğru olduğunda bazı komutları çalıştırmak, yanlış olduğunda başka komutları çalıştırmak için:

```
if (kosu1) {
    komut1;
    komut2;
} else {
    baskaKomut1;
    baskaKomut2;
} ...
}
```

- İki koşulu mantıksal VE bağlacıyla bağlamak için:

```
if (kosu1 && kosu2) {
} ...
```

VE durumundaki koşulun doğru olması için her iki alt koşulun da doğru olması gerekir.

- İki koşulu mantıksal VEYA bağlacıyla bağlamak için:

```
if (kosu1 || kosu2) {
} ...
```

VEYA durumundaki koşulun doğru olması için iki alt koşuldan birinin doğru olması yeterlidir.

- `dist(x1, y1, x2, y2)` ile `x1,y1` ve `x2,y2` noktaları arasındaki uzaklığı bulabiliriz.

Etkileşim

- `mouseClicked()` metodu içinde yazılan komutlar, fare tıklandığında çalıştırılır.
- Farenin hangi butonuna tıklandığını bulmak için `mouseButton` değişkeni kullanılır.
- İki değer birbirine eşit olup olmadığını bulmak için `==` operatörü kullanılır. Örnek:

```
if (mouseButton == LEFT) {
} ...
```

- `keyPressed()` metodu içinde yazılan komutlar, klavyede bir tuşa basıldığında çalıştırılır.
- Klavyenin hangi butonuna tıklandığını bulmak için `key` ya da `keyCode` değişkeni kullanılır. Bir harf değeri olan tuşlar için `key`, diğer tuşlar için `keyCode` kullanılır. Örnek:

```
if (keyCode == LEFT) {
    x = x - 5;
} else if (keyCode == RIGHT) {
    x = x + 5;
} else if (key == 'w') {
    y = y - 5;
} else if (key == 's') {
    y = y + 5;
}
```

Kod formatı

- Alt alta gelen komutlar aynı hizada yazılmalıdır.
- Süslü parantezlerle belirlenen kod bloklarının içi (örn. metod içleri, `if/else` içleri) iki boşluk içerde olacak şekilde yazılmalıdır.
- `Edit > Auto format` menüsü kullanılarak kodun otomatik olarak formatlanması sağlanır.

Kod içi yorumlar

- /* ve */ işaretleri arasındaki yazıları bilgisayarımız çalıştırmaz, tamamen görmezden gelir.
- // işaretlerinden itibaren satır sonuna kadar olan yazıları bilgisayarımız çalıştırmaz, tamamen görmezden gelir.

Ses

- Kodumuzun başına şu satır eklenir:

```
import ddf.minim.*;
```

- setup() ve draw() dışına, anlık çalmak istediğimiz her bir ses için şu şekilde bir değişken tanımlanır:

```
AudioSample ding;
```

- setup() ve draw() dışına, arka planda çalmak istediğimiz her bir ses için şu şekilde bir değişken tanımlanır:

```
AudioPlayer bazinga;
```

- setup() metodunun içinde, en başta, minim değişkeni tanımlanır. AudioSample ve AudioPlayer değişkenlerine müzik dosyaları yüklenir.

```
void setup() {  
  Minim minim = new Minim(this);  
  bazinga = (AudioPlayer)minim.loadFile("bazinga.mp3");  
  ding = minim.loadSample("ding.mp3"); // ses yukleme  
}
```

- AudioPlayer olarak tanımlanmış sesler loop() ile döngü şeklinde çalınır, pause() ile durdurulur.

```
bazinga.loop();  
bazinga.pause();
```

- AudioSample olarak tanımlanmış sesler trigger() ile çalınır.

```
ding.trigger();
```

Yazı

- Sağ alt köşesi 100, 200 noktasında olacak şekilde ekrana metin çizmek için:

```
text("Merhaba", 100, 200);
```

- textSize(32): Font boyutunu örn. 32 yapar.