

Übungsblatt 2- EIPNF WS 15/16

Besprechung des Übungsblatts erfolgt am 11.11. & 14.11.

Aufgabe 1:

Hier sehen wir den in der Vorlesung besprochenen Code. Aufgabe ist es die zwei Funktionen `drawAnimationA()` und `drawAnimationB()` mit Animationen zu befüllen. Zur Auswahl stehen folgende Optionen von denen 2 umgesetzt werden sollen:

1. Ein Kreis bewegt sich abhängig vom `animationCounter` von links nach rechts über den Bildschirm. (Dimensionen beliebig)
2. Ein Kreis schrumpft von einem maximalen Wert zu 0. (Vorsicht vor negativen Werten)
3. Der Hintergrund wechselt zufällig die Farben. (Tipp: eine Variable des Typs `color` ist hilfreich die in der Animation verändert wird)
4. Ein Rechteck wird gezeichnet, dessen Position bei jedem erneuten Tastendruck wechselt aber bei gehaltener Taste am selben Ort bleibt.

```
int animationCounter;

void setup () {
  size(600, 400);
  animationCounter = 0;
}

void draw () {
  background(0);
  handleInput();
  drawAnimationA();
  //drawAnimationB();
}

void handleInput () {
  if (keyPressed) {
    animationCounter++;
  } else {
    animationCounter = 0;
  }
}

void drawAnimationA () {
  if (animationCounter>0) { // hier kann deine Animation stattfinden
  }
}

void drawAnimationB () {
  if (animationCounter>0) { // hier kann deine Animation stattfinden
  }
}
```

Aufgabe 2:

Nun sollen die eben erarbeiteten Animationen einmal bei Maustastendruck und einmal bei Keyboardtastendruck aktiviert werden. Dazu benötigen wir natürlich zwei `animationCounter` und mehrere Funktionen die unsere Funktionalität erweitern.

```
// je ein animationCounter für Mausinput und Tasteninput

void setup () {
  size(600, 400);
  // initialisiere beide animationCounter
}

void draw () {
  background(0);

  // handleMouseInput
  // handleKeyInput

  // drawMouseAnimation
  // drawKeyAnimation
}

void handleMouseInput () {
  // ...
}

void handleKeyInput () {
  // ...
}

void drawMouseAnimation () {
  // ...
}

void drawKeyAnimation () {
  // ...
}
```