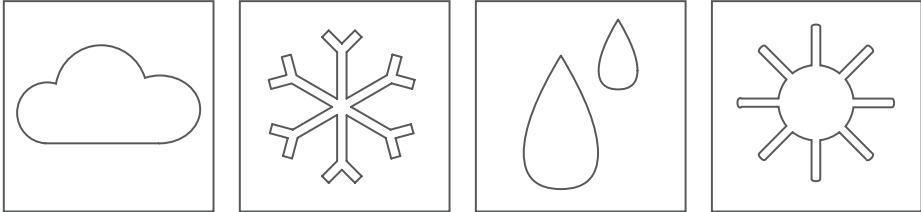


umgangsFormen weatherPix

die Idee
weatherPix ist eine Garderobe, die einem die aktuelle Wetterlage anzeigt. Die Garderobe ist neben dem Kleiderschrank der Ort in der Wohnung, an dem die Frage nach dem aktuellen Wetter interessant ist, denn die Antwort entscheidet über die Wahl der Kleidung oder über die Frage, ob es sinnvoll wäre, einen Schirm mitzunehmen.

das Konzept
Wetterdaten liegen als RSS- Feeds im Internet vor, z.B. bei weather.com. Ein verbreiteter Anwendungsbereich dieser Wetter- RSS Feeds besteht in Form von Widgets auf dem Computer oder auf mobilen Geräten. Hier aktualisieren sich die Wetterdaten selbstständig bzw. jedes Mal dann, wenn das Widget vom Benutzer aktiviert wird. Die Garderobe ist über WLAN mit dem Internet verbunden und in der Lage diese RSS Feeds zu lesen, um die relevanten Wetterinformationen in ein analoges Display zu übersetzen.

Einzelne Wettersymbole stellen in Kombination miteinander die jeweilige Wettersituation dar.



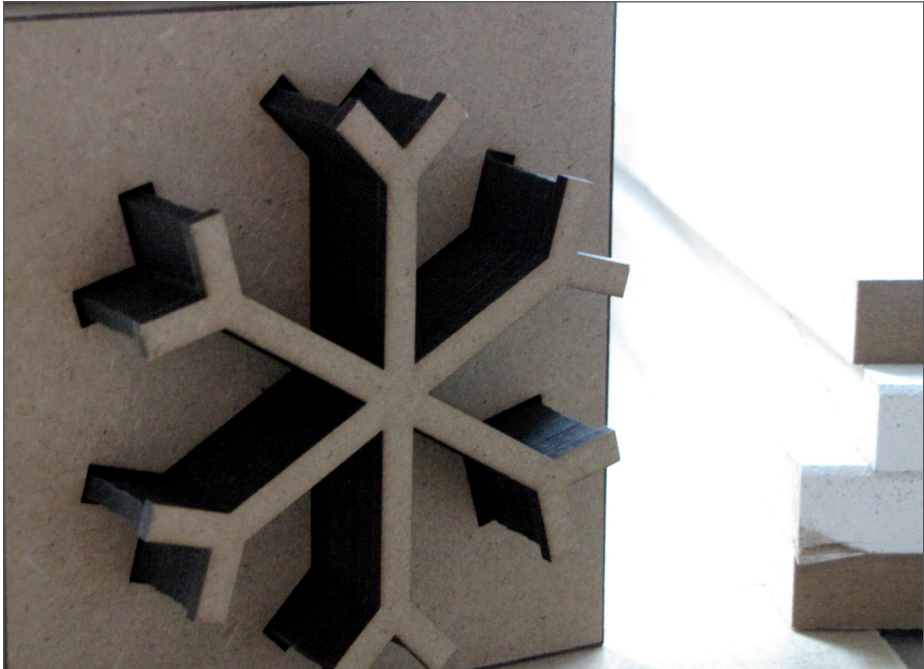
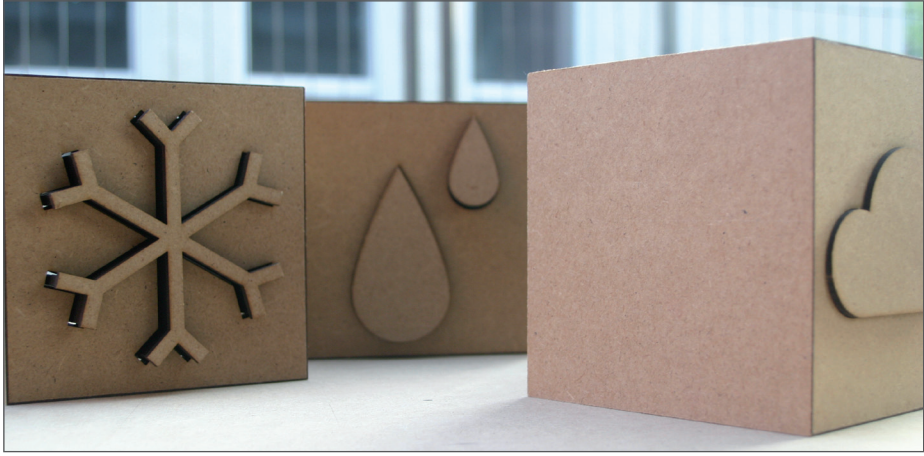
Hierbei werden die in komplexer Form vorliegenden Wetterdaten auf die für den Benutzer relevanten Informationen reduziert. Nämlich auf die Informationen, die man benötigt, um sich für das richtige Kleidungsstück oder zusätzliche Accessoires zu entscheiden.

Brauche ich einen Schirm? Schneit es noch?

Umsetzung
Für die Gestaltung des Displays wird ein modulares System aus hängenden, eigenständigen Würfeln bzw. „physischen Pixeln“, die jeweils ein Wettersymbol repräsentieren, gewählt. Pixel deshalb, da hier in digitaler Form vorliegende Informationen in ein analoges und mechanisch bewegtes Display transformiert werden. Pixel sind der visuelle Baustein unserer digitalen Welt und in diesem Fall bilden sie auch den Baustein für die Übersetzung in die physische Welt. Die Boxen agieren eigenständig und sind prinzipiell durch andere Wettersymbole erweiter- oder austauschbar.

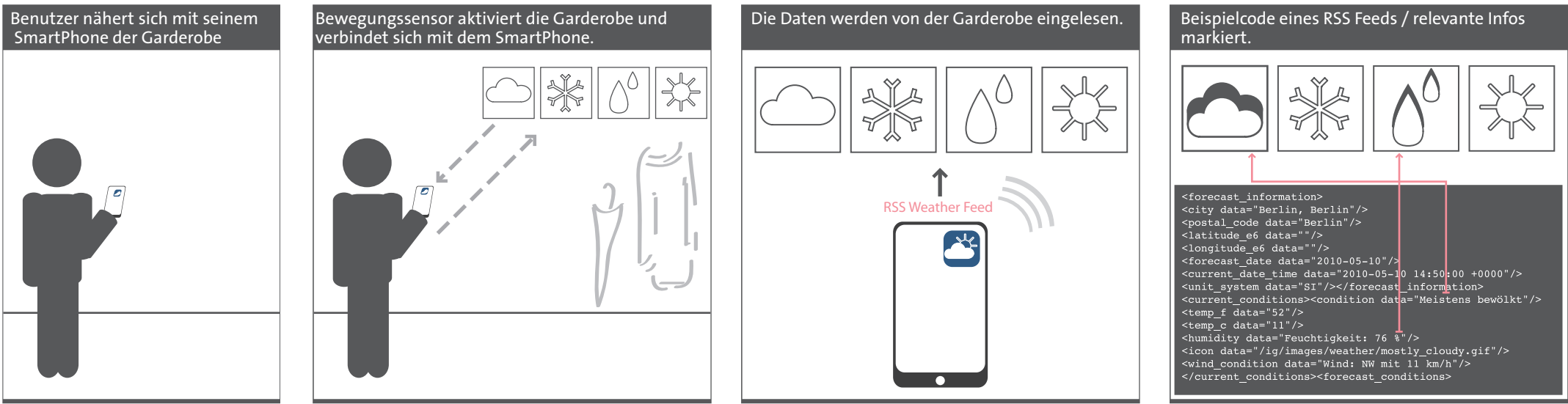
zwei mögliche Szenarien
In Variante 1 reagiert die Garderobe über einen Bewegungssensor direkt auf den Benutzer und verbindet sich mit einem Wetter RSS Feed Internet, liest die jeweiligen Daten aus und zeigt das Wetter an.

In Variante 2 reagiert die Garderobe ebenfalls über einen Bewegungssensor auf den Benutzer, will aber die aktuellen Wetterdaten über ein Widget auf dessen Smartphone auslesen. Trägt der Benutzer sein Smartphone nicht bei sich, bleibt die Garderobe passiv und erinnert ihn somit indirekt daran, dass er gerade in Begriff ist das Haus ohne Handy zu verlassen.



Die Wetter Pixel sind aus gelasertem MDF (3mm). Im Innern befindet sich ein einfacher Servo Motor, der durch ein ARDUINO Board kontrolliert wird.

Storyboard für Variante 2



Simulation
Für die Ausstellung am Tag der Offenen Tür wird dem Besucher die Möglichkeit gegeben, selbst das Wetter zu manipulieren und dadurch die Garderobe in Aktion zu erleben. Um eine inhaltliche Nähe zum Konzept herzustellen, passiert die Manipulation des Wetters durch eine Klimakarte auf dem Bildschirm, der diese Daten via PROCESSING an das Herzstück dieser Installation sendet, das ARDUINO.



```
PROCESSING Code

//PROCESSING sendet WETTER an PROCESSING
import processing.serial.*;
PImage Klimakarte;

//Städtenamen
PImage Madrid;
PImage Rom;
PImage Stockholm;
PImage London;
PImage Berlin;
PImage Moskau;
PImage Istanbul;
PImage Rejkjavik;

int value =0;
Serial port;

void setup() {
  size(1024,650);
  smooth();
  Klimakarte = loadImage("Klimakarte.png");
  Madrid = loadImage("Madrid.png");
  London = loadImage("London.png");
  Moskau = loadImage("Moskau.png");
  Rom = loadImage("Rom.png");
  Stockholm = loadImage("Stockholm.png");
  Istanbul = loadImage("Istanbul.png");
  Berlin = loadImage("Berlin.png");

  image(Klimakarte,0,0);
  println(Serial.list());
  port= new Serial(this, Serial.list()[1], 9600);
}

void mouseClicked(){
  //Weather Situation MADRID
  if (mouseX >207 && mouseX <291 && mouseY>378 && mouseY<462) {
    image(Madrid, 0,0);
    port.write('M');
    println('M');
  }

  //Weather Situation LONDON
  if (mouseX >368 && mouseX <452 && mouseY>183 && mouseY<267) {
    image(London, 0,0);
    port.write('L');
    println('L');
  }

  //Weather Situation MOSKAU
  if (mouseX >783 && mouseX <867 && mouseY>141 && mouseY<225) {
    image(Moskau, 0,0);
    port.write('K');
    println('K');
  }

  //Weather Situation STOCKHOLM
  if (mouseX >580 && mouseX <664 && mouseY>99 && mouseY<183) {
    image(Stockholm, 0,0);
    port.write('S');
    println('S');
  }

  //Weather Situation PARIS
  if (mouseX >384 && mouseX <468 && mouseY>294 && mouseY<378) {
    image(Paris, 0,0);
    port.write('P');
    println('P');
  }

  //Weather Situation BERLIN
  if (mouseX >526 && mouseX <600 && mouseY>225 && mouseY<311) {
    image(Berlin, 0,0);
    port.write('B');
    println('B');
  }

  //Weather Situation ISTANBUL
  if (mouseX >767 && mouseX <851 && mouseY>404 && mouseY<488) {
    image(Istanbul, 0,0);
    port.write('U');
    println('U');
  }

  //Weather Situation ROM
  if (mouseX >484 && mouseX <568 && mouseY>394 && mouseY<478) {
    image(Rom, 0,0);
    port.write('O');
    println('O');
  }

  // Mouse moving over the Image
  void mouseMoved(){
    if (mouseX >207 && mouseX <291 && mouseY>378 && mouseY<462) {
      image(Madrid, 0,0);
    }
    if (mouseX >368 && mouseX <452 && mouseY>183 && mouseY<267) {
      image(London, 0,0);
    }
    if (mouseX >783 && mouseX <867 && mouseY>141 && mouseY<225) {
      image(Moskau, 0,0);
    }
    if (mouseX >580 && mouseX <664 && mouseY>99 && mouseY<183) {
      image(Stockholm, 0,0);
    }
    if (mouseX >384 && mouseX <468 && mouseY>294 && mouseY<378) {
      image(Paris, 0,0);
    }
    if (mouseX >526 && mouseX <600 && mouseY>225 && mouseY<311) {
      image(Berlin, 0,0);
    }
    if (mouseX >767 && mouseX <851 && mouseY>404 && mouseY<488) {
      image(Istanbul, 0,0);
    }
    if (mouseX >484 && mouseX <568 && mouseY>394 && mouseY<478) {
      image(Rom, 0,0);
    }
  }

  void keyPressed(){
    image(Klimakarte, 0, 0);
    port.write('Q');
    println('Q');
  }
}
```

```
ARDUINO Code

//ARDUINO liest WETTER-INFO aus PROCESSING

#include <Servo.h>
Servo myServoRain;
Servo myServoCloud;
Servo myServoSun;
Servo myServoSnow;

int incomingByte;
void setup() {
  Serial.begin(9600);

  myServoRain.attach(9);
  myServoCloud.attach(10);
  myServoSun.attach(11);
  myServoSnow.attach(3);

  myServoRain.write(0);
  myServoCloud.write(0);
  myServoSun.write(0);
  myServoSnow.write(0);

  void loop() {
    if (Serial.available() > 0) {
      incomingByte = Serial.read();
      //London
      if (incomingByte == 'L') {
        resetServos();
        myServoRain.write(-60);
        myServoCloud.write(-50);
      }
      //Madrid
      if (incomingByte == 'M') {
        resetServos();
        myServoSun.write(-80);
        myServoCloud.write(-40);
      }
      //Moskau
      if (incomingByte == 'K') {
        resetServos();
        myServoSun.write(0);
        myServoCloud.write(0);
      }
      //Paris
      if (incomingByte == 'P') {
        resetServos();
        myServoSun.write(0);
        myServoCloud.write(-50);
      }
      //Istanbul
      if (incomingByte == 'U') {
        resetServos();
        myServoSun.write(20);
        myServoRain.write(10);
      }
      //ROM
      if (incomingByte == 'O') {
        resetServos();
        myServoSun.write(30);
        myServoRain.write(60);
      }
      //Stockholm
      if (incomingByte == 'S') {
        resetServos();
        myServoCloud.write(50);
        myServoRain.write(20);
        myServoSun.write(10);
      }
      //Berlin
      if (incomingByte == 'B') {
        resetServos();
        myServoCloud.write(40);
        myServoSun.write(10);
      }
      if (incomingByte == 'Q') {
        resetServos();
      }
    }
    delay(1000);
  }

  void resetServos(){
    myServoRain.write(110);
    myServoCloud.write(40);
    myServoSun.write(70);
    myServoSnow.write(50);
  }
}
```

