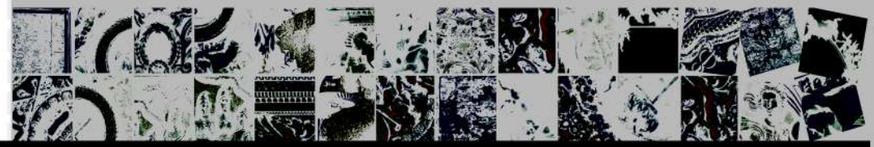


CABINET OF INFINITY

Wunderkammer 4.0 / Cabinets of wonder“
Schloss Friedenstein Gotha.

Author : Zeinab Rahimi , Faezeh Mansourkhaki

Media Architecture
Winter2022

Cabinet Of Infinity

„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold,
Nezar Abuhalaweh
Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

About The Project:

The fundamental concept of our design was to refer to baroque architecture in terms of highly decorative interior space in a new and contemporary way.
We decided to use the minimal surface equation in grasshopper to develop our geometry models. These forms are going to work as a media environment to show some photo collections and also as an interactive place for the visitors.
The software Touchdesinger, the Arduino Uno, and gesture sensor is used as media technology to make the space interactive.
During the design process, we developed 4 modules that are related to the chosen collection of photos, sculptures, paintings, and objects of Gotha museums.

„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

Cube inspired by Baroque Cabinet of Wonder

The main idia of the cube is as fallows**:**

- Layered architecture of baroque style that is represented by transparent papers layers.
-Movement as one of the important aspects of baroque architecture is represented by paper positions(how to place the papers)
-The word baroque itself means unconventional; a collage of screenshots that seem pointless with museum objects was used to represent this concept.
-The harmony and norm among various and intertwined elements
and themes is shown by creating a collage with the concept of two irrelevant images
-Light as an important component in Baroque paintings and images is used.

- Layered architecture
-Movement
- Unconventional collage of screenshots
-The harmony and norm among various and intertwined elements and themes
-Light

Individual Projrct

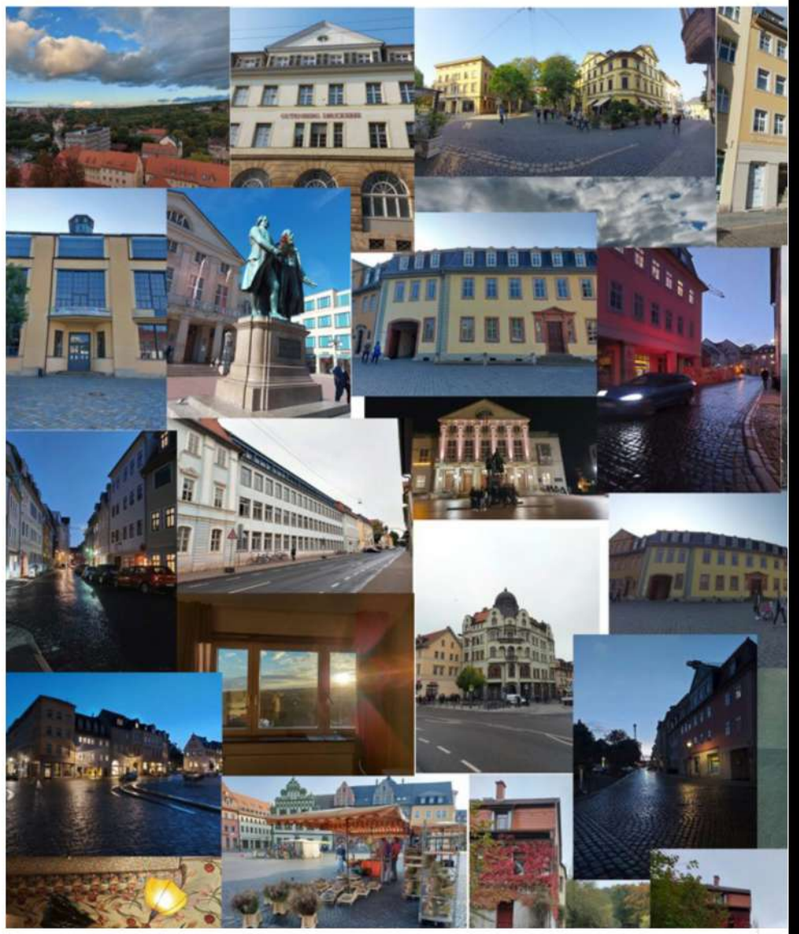
First Plenum

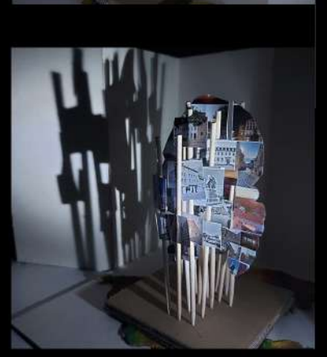
PAGE 1

Introductory Project

Cube inspired by Baroque Cabinet of Wonder

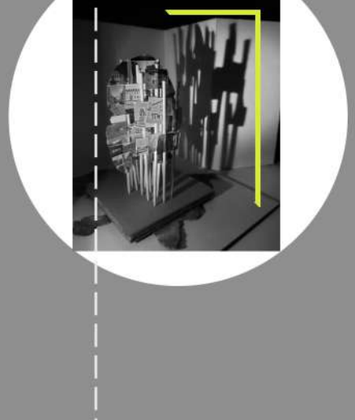
The personal interpretation of an art piece is different for each visitor.
The visitor's background and experiences and memories make each person's personal perception u n i q u e .
This cube shows the collection of photos that I took during my stay in Weimar.
In this cube, the light represents the visitor's eye. The shadow behind represents the individual perception of each visitor.
By changing the position of light (different visitors point of view) the shadow behind changes accordingly.
Since the most strong memory from Weimar could be the "Johann Wolfgang von Goethe", the pictures have been arranged in a way that the central shadow behind the collection shows a famouse





Constant movement and change of the shadow behind the photo collection (art piece) represent a uniquely personal interpretation of each visitor.



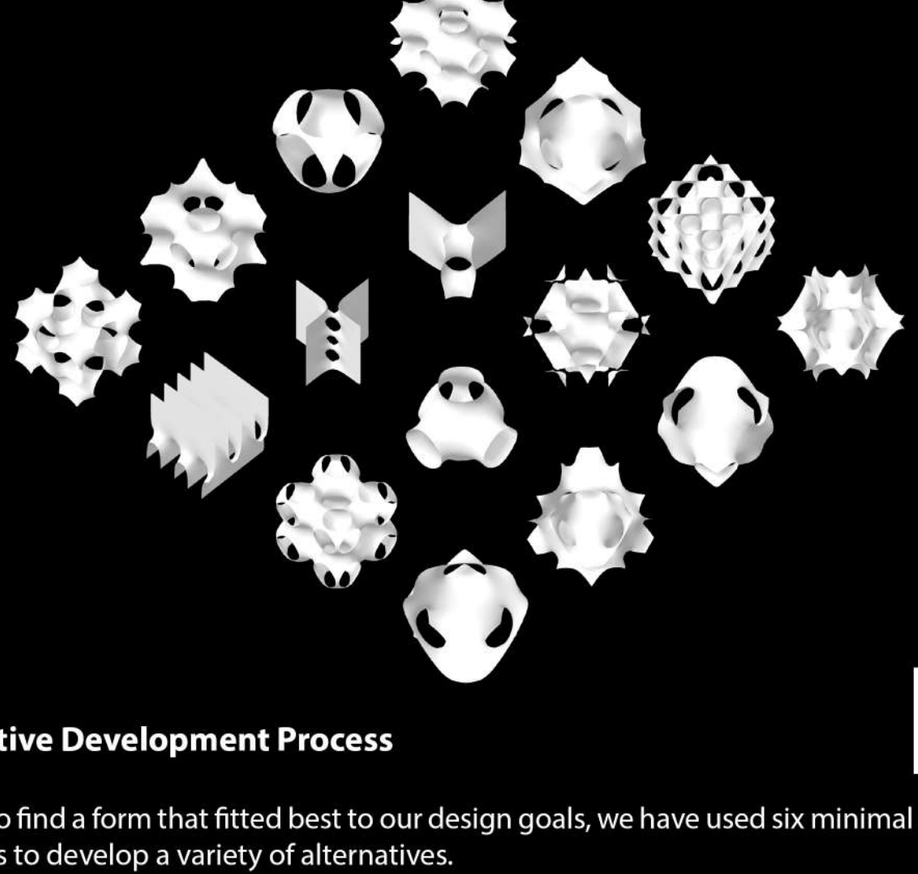
Light ——— Shadow
Movement
Layers

- The common concepts between the two Cubes can be summarised as follows :

- The layers of pictures
-The element of movement and constant change
- The Light as a source
- The shadow that is a non-seprable entity from light

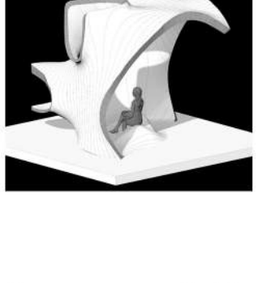
„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh
Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

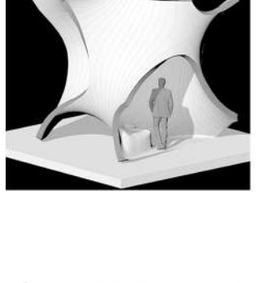


Alternative Development Process

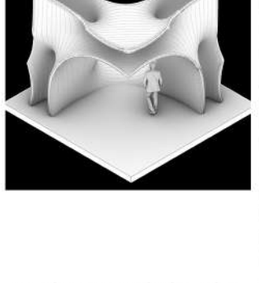
In order to find a form that fitted best to our design goals, we have used six minimal surface equations to develop a variety of alternatives.
Four modules which both aesthetically and functionally could fulfill the design, were finally chosen.



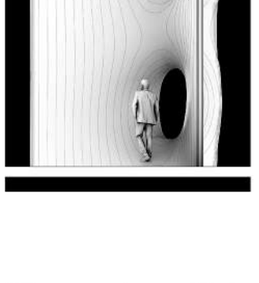
A new interpretation of the **"Object"** collection of the Gotha museums in form of collage pictures is shown in this module.



A contemporary collage made of the **“SCULPTURES”** of Gotha musumes collection is shown in this module.



In this module, the visitor can enjoy watching the valuable **“PAINTINGS”** of Gotha musumes in a new way.

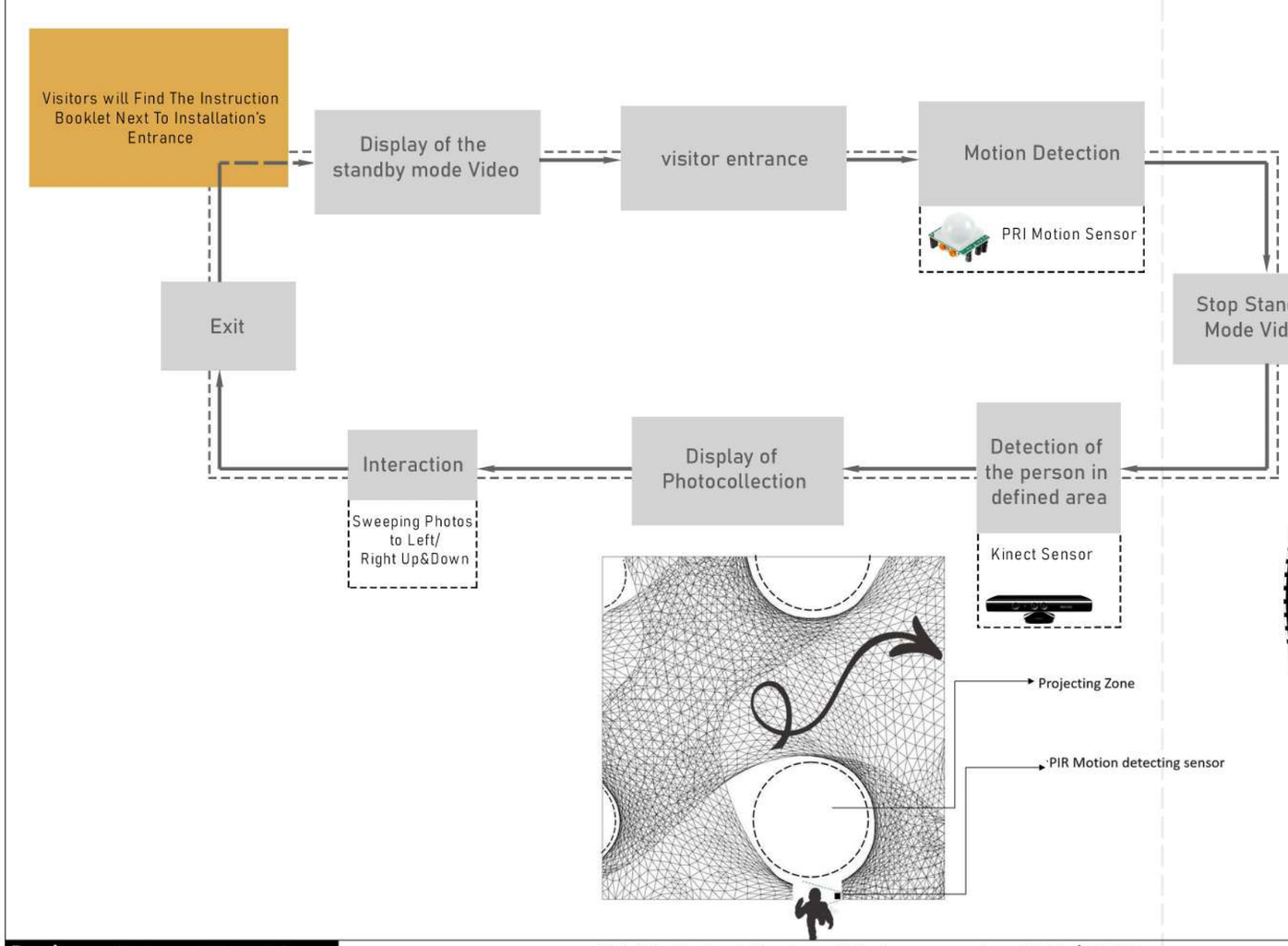


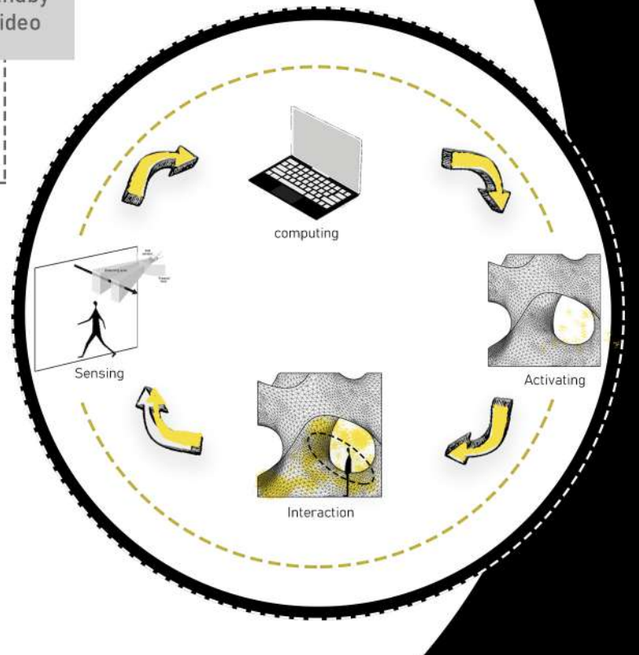
“The beautiful **“PATTERNS”** and ceilings of Gotha museums are shown to the visitors in this module.

„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki





In order to make the space interactive, a gesture sensor (for the prototype) is used. In the real project, Kinect cameras would be a more appropriate technology to be used.

A standby mode picture that mainly emphasizes the beauty of the form is displayed by default, when a visitor enters the space the motion detector detects the motion and the standby mode display would be stopped after certain minutes, then the presence of the person would be detected by the Kinect camera and in the next step, a photo collection would be displayed on the interior surface.

the Kinect camera gives the visitor the freedom to sweep the photos in different directions(right-left-up-down). the visitor also is able to zoom in and zoom out the photos. the main concept for this interaction is to give a visitor a sense of control over the displayed photos.

„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

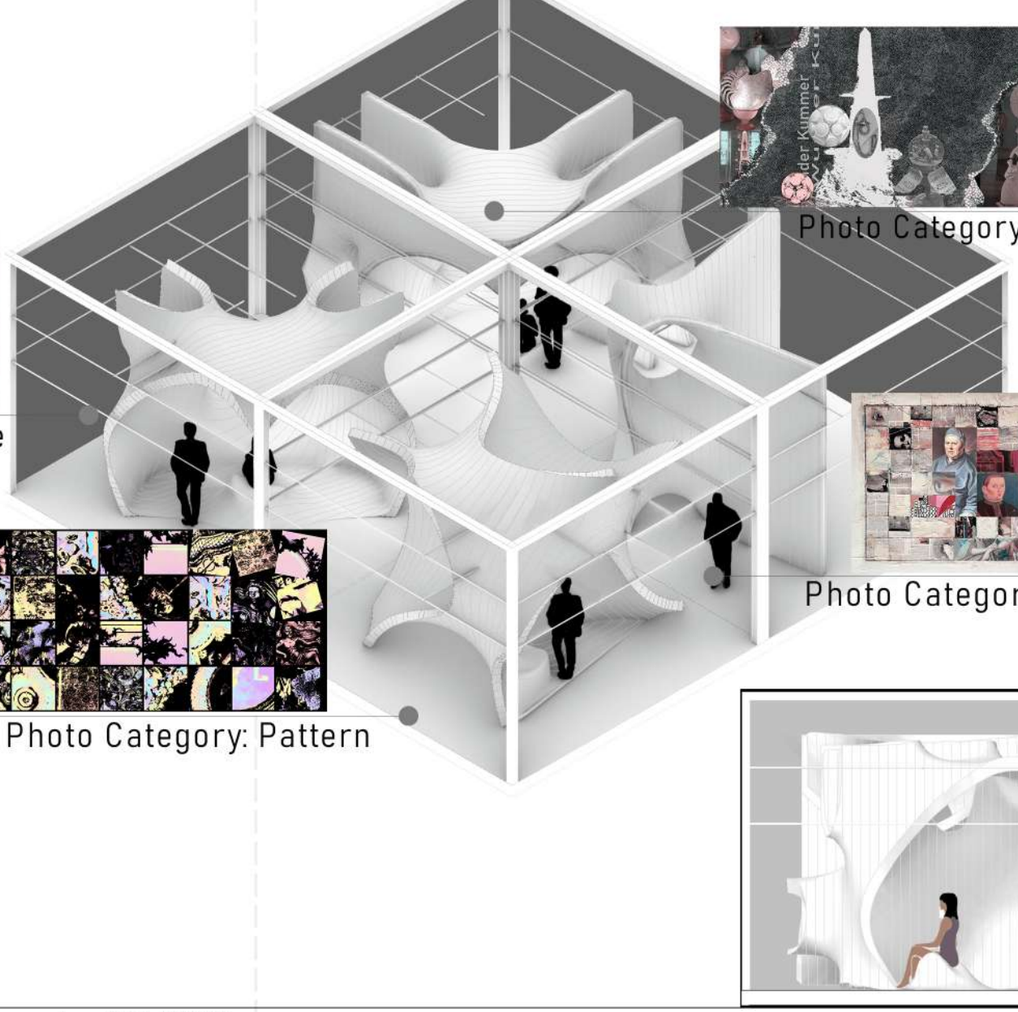
Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

The fundamental design concepts of this two-layer pavilion is based on the baroque architecture concept of high decorative interior design and relatively simpler faced.
The intention is to show pictures of art pieces through interactive LED screens and give the visitor the joy of moving the displayed photo collection by themselves.
we have chosen four modules to show four categories of art pieces which are: Sculpture, Paintings, Patterns, and Objects.

Photo Category: Sculpture

Photo Category: Objects



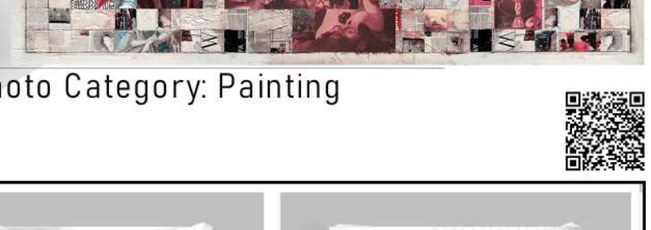


Photo Category: Painting

Photo Category: Pattern

„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

Touch Designer inner shell code

```
# me - this DAT
#
# dat - the DAT that received the data
# rowIndex - the row number the data was placed into
# message - an ascii representation of the data
#           Unprintable characters and unicode characters will
#           not be preserved. Use the 'bytes' parameter to get
#           the raw bytes that were sent.
# bytes - byte array of the data received

def onReceive(dat, rowIndex, message, bytes):
    #message = op('select1')[0,0]
    #print(message)
    if message == 'RIGHT':
        op('horiz').par.value0 = 1
    elif message == "LEFT":
        op('horiz').par.value0 = -1
    elif message == "UP":
        op('vert').par.value0 = 1
    elif message == "DOWN":
        op('vert').par.value0 =-1
    elif message == "NEAR":
        op('zoom').par.value0 = 1
    elif message == "FAR":
        op('zoom').par.value0 = -1
    elif message == "NONE":
        op('horiz').par.value0 = 0
        op('vert').par.value0 = 0
        op('zoom').par.value0 = 0
    return
```

The code to connect the arduino to Touchdesinger and convert the input data (from sensor) to move the photo on geometry

RESET

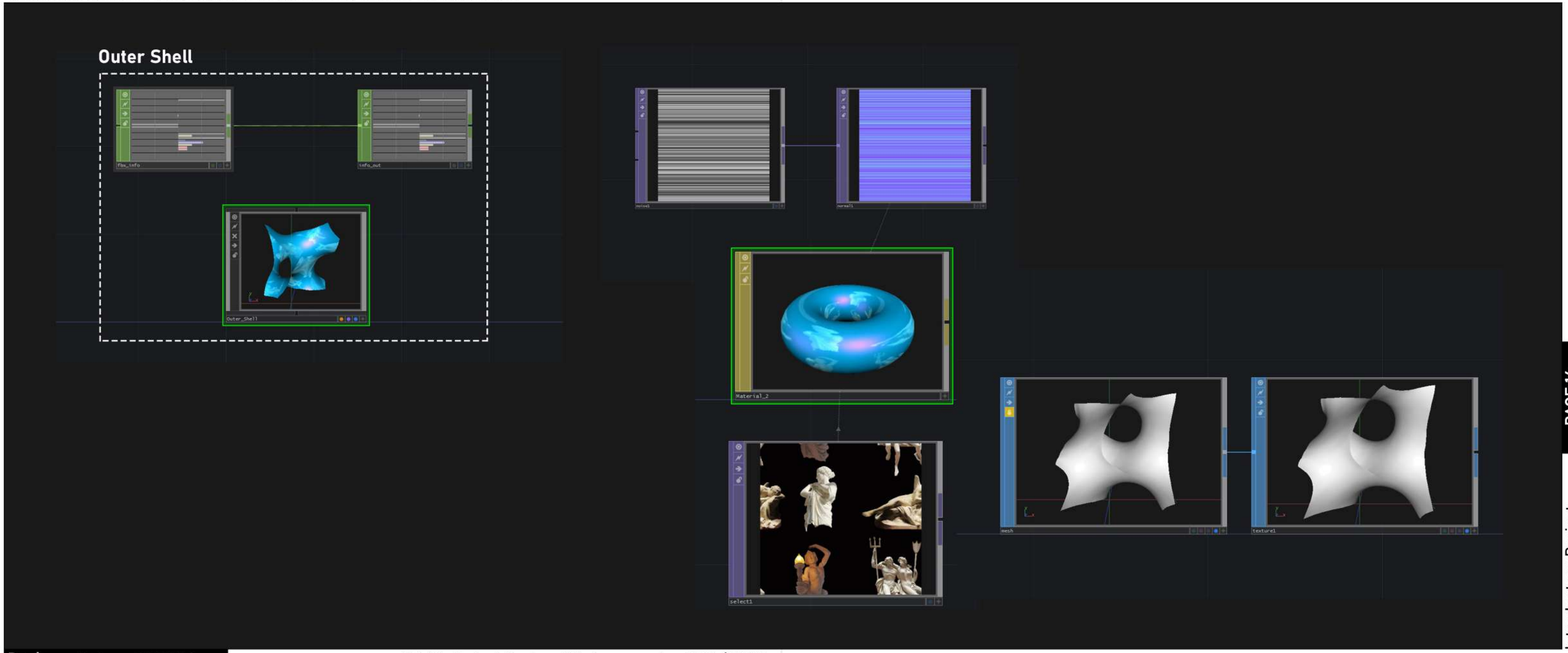
```
# me - this DAT
#
# channel - the Channel object which has changed
# sampleIndex - the index of the changed sample
# val - the numeric value of the changed sample
# prev - the previous sample value
#
# Make sure the corresponding toggle is enabled in the CHOP Execute DAT.

def onOffToOn(channel, sampleIndex, val, prev):
    op('speed1').par.resetpulse.pulse()
    op('speed2').par.resetpulse.pulse()
    op('speed3').par.resetpulse.pulse()
    return

def whileOn(channel, sampleIndex, val, prev):
    return

def onOnToOff(channel, sampleIndex, val, prev):
    return

def whileOff(channel, sampleIndex, val, prev):
    return

def onValueChange(channel, sampleIndex, val, prev):
    return
```

The code used to reset the data

„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

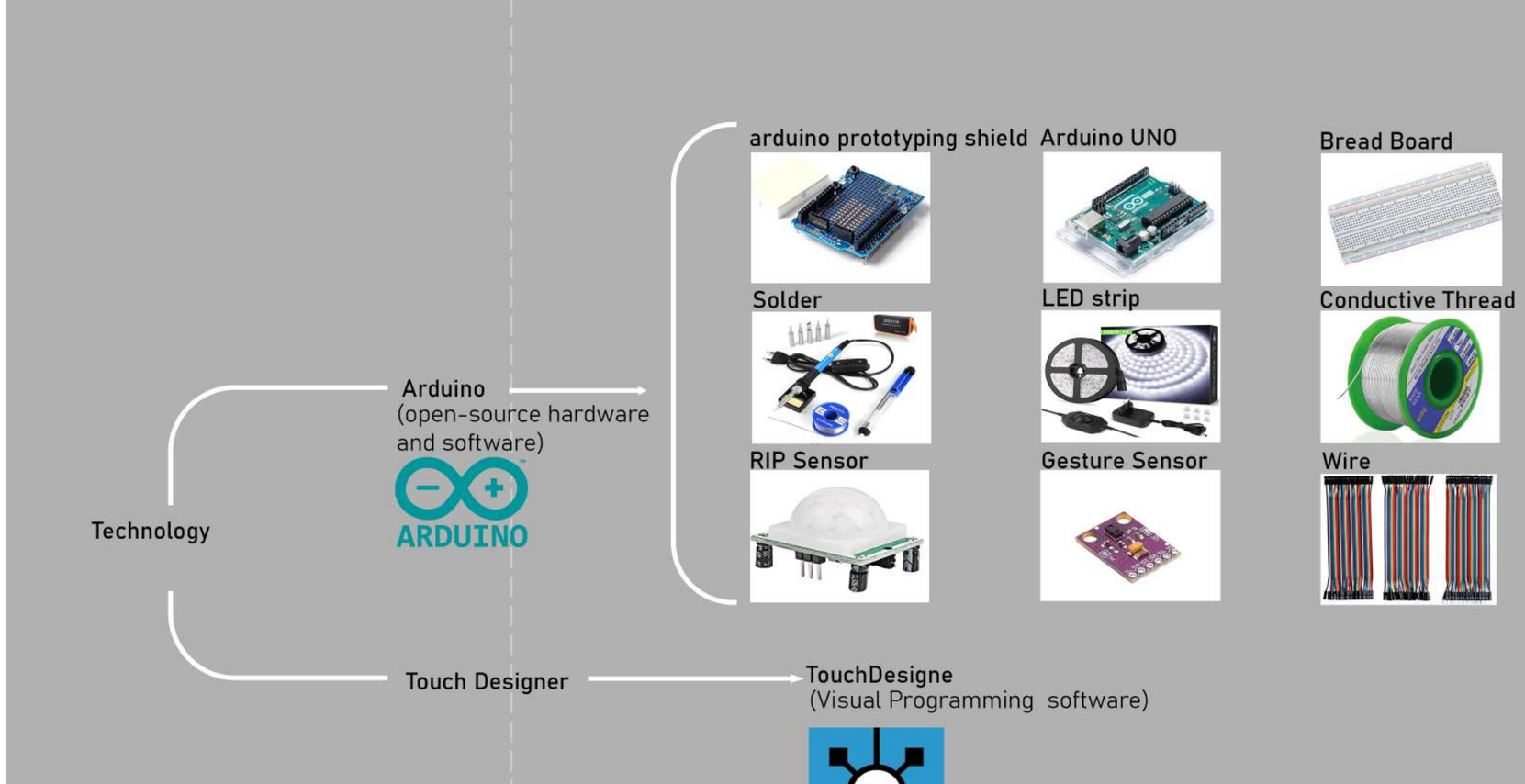


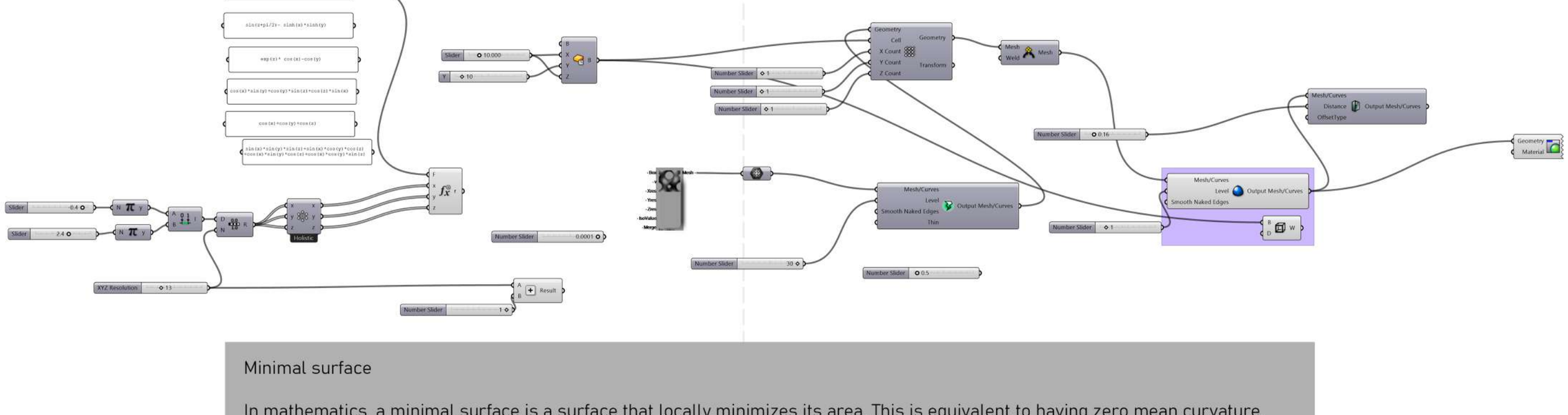
„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

Technology


Technology
Arduino
(open-source hardware and software)
ARDUINO
arduino prototyping shield
Arduino UNO
Bread Board
Solder
LED strip
Conductive Thread
RIP Sensor
Gesture Sensor
Wire
Touch Designer
TouchDesigne
(Visual Programming software)

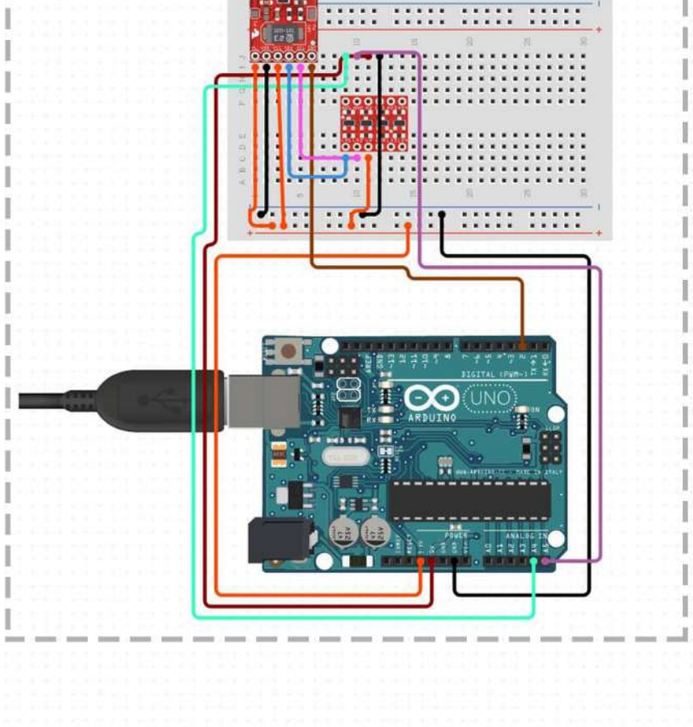


Minimal surface

In mathematics, a minimal surface is a surface that locally minimizes its area. This is equivalent to having zero mean curvature.
The term "minimal surface" is used because these surfaces originally arose as surfaces that minimized total surface area subject to some constraint.
Minimal surfaces can be defined in several equivalent ways in R3. The fact that they are equivalent serves to demonstrate how minimal surface theory lies at the crossroads of several mathematical disciplines, especially differential geometry, calculus of variations, potential theory, complex analysis and mathematical physics.

Source :
https://en.wikipedia.org/wiki/Minimal_surface

```ino
#include <Wire.h>
#include <SparkFun_APDS9960.h>

// Pins
#define APDS9960_INT    2 // Needs to be an interrupt pin

// Constants

// Global Variables
SparkFun_APDS9960 apds = SparkFun_APDS9960();
int isr_flag = 0;

void setup() {

  // Set interrupt pin as input
  pinMode(APDS9960_INT, INPUT);

  // Initialize Serial port
  Serial.begin(9600);
  Serial.println();
  Serial.println(F("--------------------------------"));
  Serial.println(F("SparkFun APDS-9960 - GestureTest"));
  Serial.println(F("--------------------------------"));

  // Initialize interrupt service routine
  attachInterrupt(0, interruptRoutine, FALLING);

  // Initialize APDS-9960 (configure I2C and initial values)
  if ( apds.init() ) {
    Serial.println(F("APDS-9960 initialization complete"));
  } else {
    Serial.println(F("Something went wrong during APDS-9960 init!"));
  }

  // Start running the APDS-9960 gesture sensor engine
  if ( apds.enableGestureSensor(true) ) {
    Serial.println(F("Gesture sensor is now running"));
  } else {
    Serial.println(F("Something went wrong during gesture sensor init!"));
  }
}
void loop() {
  if( isr_flag == 1 ) {
    detachInterrupt(0);
    handleGesture();
    isr_flag = 0;
    attachInterrupt(0, interruptRoutine, FALLING);
  }
}

void interruptRoutine() {
  isr_flag = 1;
}

void handleGesture() {
    if ( apds.isGestureAvailable() ) {
    switch ( apds.readGesture() ) {
      case DIR_UP:
        Serial.println("UP");
        break;
      case DIR_DOWN:
        Serial.println("DOWN");
        break;
      case DIR_LEFT:
        Serial.println("LEFT");
        break;
      case DIR_RIGHT:
        Serial.println("RIGHT");
        break;
      case DIR_NEAR:
        Serial.println("NEAR");
        break;
      case DIR_FAR:
        Serial.println("FAR");
        break;
      default:
        Serial.println("NONE");
    }
  }
```

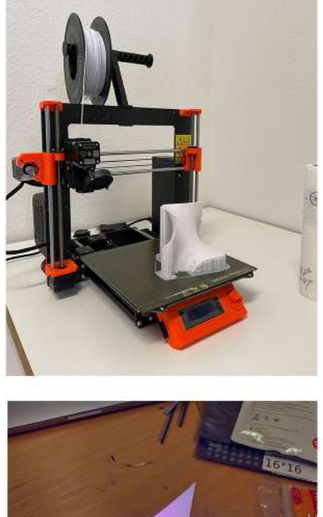
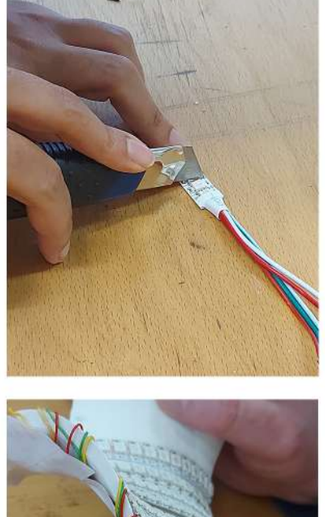
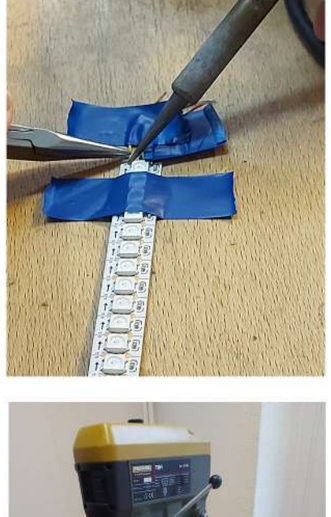
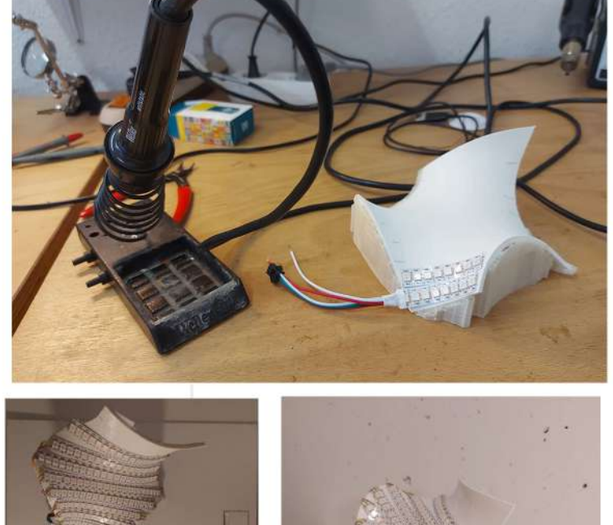
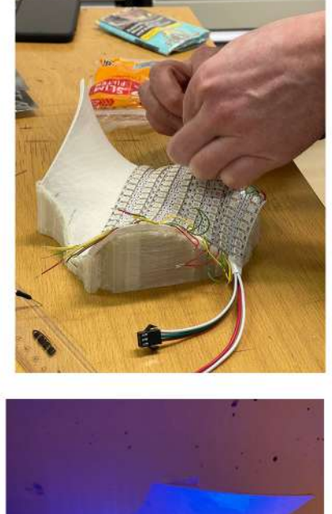
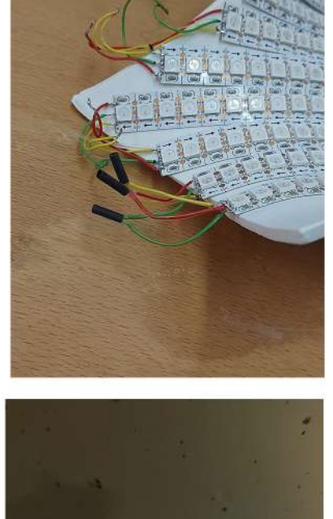


„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Ze

physical prototype procedure

physical prototyping arduino code

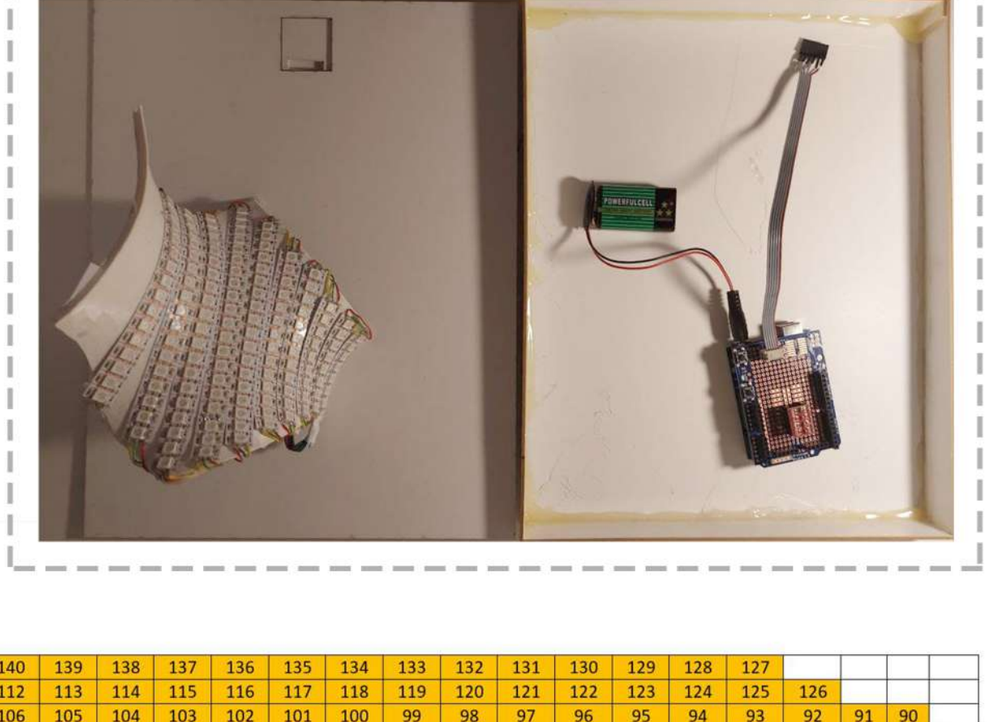
```
//Neopixel stuff
#include <Adafruit_GFX.h>
#include <Adafruit_NeoMatrix.h>
#include <Adafruit_NeoPixel.h>
#ifndef PSTR
 #define PSTR // Make Arduino Due happy
#endif

#define PIN 6

//Gesture sensor stuff
#include <Wire.h>
#include <SparkFun_APDS9960.h>

// Pins
#define APDS9960_INT    2 // Needs to be an interrupt pin

// Constants

// Global Variables
SparkFun_APDS9960 apds = SparkFun_APDS9960();
int isr_flag = 0;
// end of Gesture sensor stuff

// Setting up the matrix
Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(21, 10, PIN, 0, NEO_GRB + NEO_KHZ800);

const uint16_t colors[] = {
  matrix.Color(255, 0, 0), matrix.Color(0, 255, 0), matrix.Color(0, 0, 255) };

const int16_t YourGrid[] = {
    -1,-1,141,140,139,138,137,136,135,134,133,132,131,130,129,128,127,-1,-1,-1,-1,
    -1,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,-1,-1,-1,
    109,108,107,106,105,104,103,102,101,100,99,98,97,96,95,94,93,92,91,90,-1,
    69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,
    -1,68,67,66,65,64,63,62,61,60,59,58,57,56,55,54,53,52,51,-1,-1,
    -1,-1,37,38,39,40,41,42,43,44,45,46,47,48,49,50,-1,-1,-1,-1,-1,
    -1,-1,-1,36,35,34,33,32,31,30,29,28,27,26,25,-1,-1,-1,-1,-1,-1,
    -1,-1,-1,-1,-1,16,17,18,19,20,21,22,23,24,-1,-1,-1,-1,-1,-1,-1,
    -1,-1,-1,-1,-1,15,14,13,12,11,10,9,8,-1,-1,-1,-1,-1,-1,-1,-1,
    -1,-1,-1,-1,0,1,2,3,4,5,6,7,-1,-1,-1,-1,-1,-1,-1,-1,-1
 };


void setup() {

  // setting up the gesture sensor
  if( isr_flag == 1 ) {
    detachInterrupt(0);
    handleGesture();
    isr_flag = 0;
    attachInterrupt(0, interruptRoutine, FALLING);
  }

// setting up the matrix
matrix.begin();
matrix.setTextWrap(false);
matrix.setBrightness(40);  // brightness
matrix.setTextColor(colors[0]);
matrix.setRemapFunction(circleGrid);

// Gesture sensor
// Initialize interrupt service routine
attachInterrupt(0, interruptRoutine, FALLING);

// Initialize APDS-9960 (configure I2C and initial values)
if ( apds.init() ) {
  Serial.println(F("APDS-9960 initialization complete"));
} else {
  Serial.println(F("Something went wrong during APDS-9960 init!"));
}

// Start running the APDS-9960 gesture sensor engine
if ( apds.enableGestureSensor(true) ) {
  Serial.println(F("Gesture sensor is now running"));
  else {
  Serial.println(F("Something went wrong during gesture sensor init!"));
  Serial.begin(9600);
  Serial.println();
  Serial.println(F("--------------------------------"));
  Serial.println(F("SparkFun APDS-9960 - GestureTest"));
  Serial.println(F("--------------------------------"));
}

int x    = matrix.width();
int pass = 0;

void loop() {
  //matrix.fillScreen(0);
    if( isr_flag == 1 ) {
    detachInterrupt(0);
    handleGesture();
    isr_flag = 0;
    attachInterrupt(0, interruptRoutine, FALLING);
  }
}

 // Brute force address conversion, XY to live pixels
 uint16_t circleGrid(uint16_t x, uint16_t y) {
    y = y * 21 + x;  // Compute index number based on 21x10 matrix
    return YourGrid[y];
}
```

physical prototyping arduino code

```
void interruptRoutine() {
  isr_flag = 1;
}

void handleGesture() {
    if ( apds.isGestureAvailable() ) {
    switch ( apds.readGesture() ) {
      case DIR_UP:
        Serial.println("UP 2");
        effect2();
        break;
      case DIR_DOWN:
        Serial.println("DOWN 1");
        effect();
        break;
      case DIR_LEFT:
        Serial.println("LEFT 3");
        effect3();
        break;
      case DIR_RIGHT:
        Serial.println("RIGHT 4");
        effect4();
        break;
      case DIR_NEAR:
        Serial.println("NEAR");
        break;
      case DIR_FAR:
        Serial.println("FAR");
        break;
      default:
        Serial.println("NONE");
    }
  }
}

void effect() {
  for(int k=0; k<3; k++) {
    for(int j=0; j<=10; j++) {  // height
      for(int i=0; i<=21; i++) {  // width
        if (k == 0) {
          matrix.drawPixel(i,j, matrix.Color(255, 0, 0));
        }
        else if (k == 1) {
          matrix.drawPixel(i,j, matrix.Color(0, 255, 0));
        }
        else {
          matrix.drawPixel(i,j, matrix.Color(0, 0, 255));
        }
      }
      matrix.show();
      delay(30);
    }
  }
}

void effect2() {
  for(int k=0; k<3; k++) {
    for(int j=10; j>=0; j--) {  // height
      for(int i=21; i>=0; i--) {  // width
        if (k == 0) {
          matrix.drawPixel(i,j, matrix.Color(255, 0, 0));
        }
        else if (k == 1) {
          matrix.drawPixel(i,j, matrix.Color(0, 255, 0));
        }
        else {
          matrix.drawPixel(i,j, matrix.Color(0, 0, 255));
        }
      }
      matrix.show();
      delay(30);
    }
  }
}

 void effect3() {
   for(int k=0; k<3; k++) {
     for(int i=21; i>=0; i--) {  // width
       for(int j=10; j>=0; j--) {  // height
         if (k == 0) {
           matrix.drawPixel(i,j, matrix.Color(255, 0, 0));
         }
         else if (k == 1) {
           matrix.drawPixel(i,j, matrix.Color(0, 255, 0));
         }
         else {
           matrix.drawPixel(i,j, matrix.Color(0, 0, 255));
         }
       }
       matrix.show();
       delay(30);
     }
   }
 }
```

physical prototyping arduino code

```
void effect4() {
  for(int k=0; k<3; k++) {
    for(int i=0; i<=21; i++) {  // width
      for(int j=0; j<=10; j++) {  // height
        if (k == 0) {
          matrix.drawPixel(i,j, matrix.Color(255, 0, 0));
        }
        else if (k == 1) {
          matrix.drawPixel(i,j, matrix.Color(0, 255, 0));
        }
        else {
          matrix.drawPixel(i,j, matrix.Color(0, 0, 255));
        }
      }
      matrix.show();
      delay(30);
    }
  }
}
```



		141	140	139	138	137	136	135	134	133	132	131	130	129	128	127				
	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126			
109	108	107	106	105	104	103	102	101	100	99	98	97	96	95	94	93	92	91	90	
69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89
	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51		
		37	38	39	40	41	42	43	44	45	46	47	48	49	50					
			36	35	34	33	32	31	30	29	28	27	26	25						
					16	17	18	19	20	21	22	23	24							
					15	14	13	12	11	10	9	8								
				0	1	2	3	4	5	6	7									

„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

First collection- sculpture

„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

Second collection- Paintings

„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

Third collection- Objects

„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

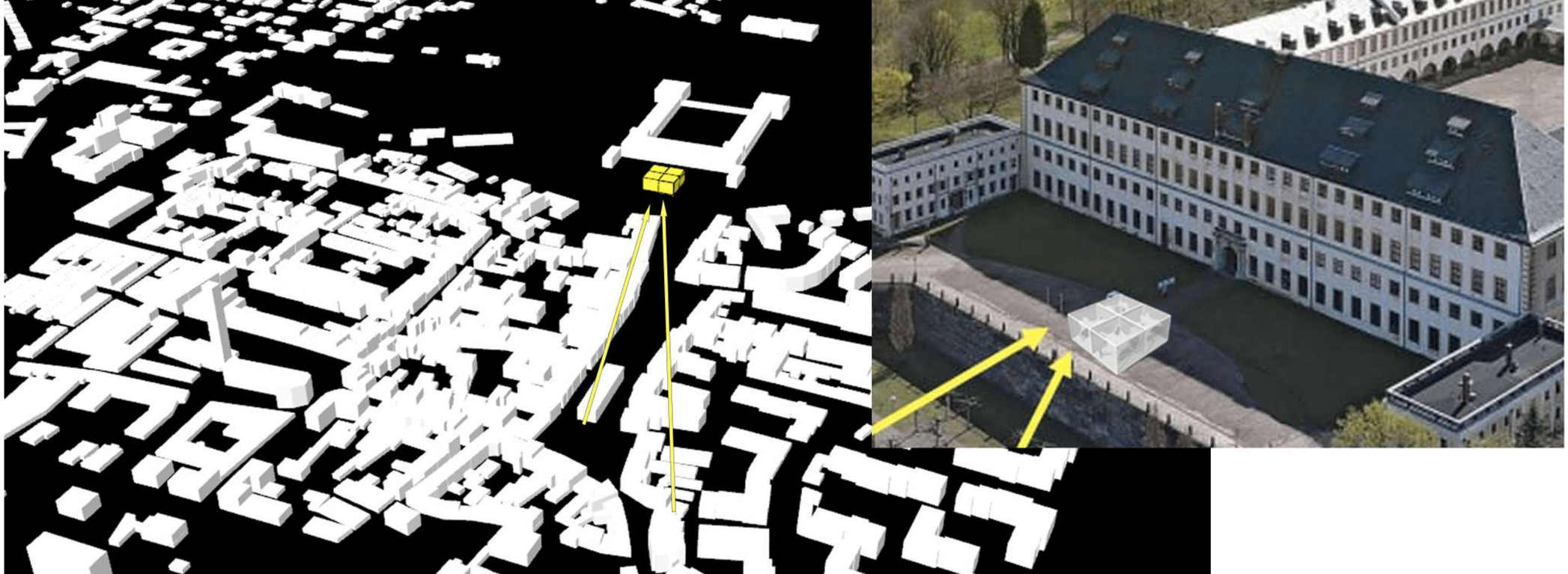
Fourth collection- Pattern

„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha

Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh

Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

Site Plan



„Wunderkammer 4.0 / Cabinets of wonder“ Schloss Friedenstein Gotha
Betreuung: Prof. Bernd Rudolf, Prof. Andreas Kästner, Junior Prof. Reinhard König, Dr. Sabine Zierold, Nezar Abuhalaweh
Gastkritik: Stefan Kraus / Author: Zeinab Rahimi, Faezeh Mansourkhaki

THANK YOU!