

BezierControl

with LeapMotion

A project of Creative coding





An interactive piece to display a bezier curve controlled by using leap motion. A project made for human-computer interaction with motion and gestures of hands.

In this project user is anyone from the academy of art university. User is controlling a bezier curve to rate the life at AAU of a particular user. After the input is done, the user will swipe hand right to left direction to save the result. Here, the result is being saved into a JSON data file.

Tasks

- Accept input from user from leapmotion device.
- save result in to JSON file to keep the data.
- Display the saved data on the screen.

Features

- use of leap motion
- based on java processing library.
- cross platform.



Development Technologies



leapmotion library integration.

```
import de.voidplus.leapmotion.*;

LeapMotion leap;
float fist = 0;
boolean pitchStatus = false;

void setup(){
    leap = new LeapMotion(this).allowGestures();
}

void leapOnSwipeGesture(SwipeGesture g, int state){
    int id = g.getId();
    Finger finger = g.getFinger();
    PVector position = g.getPosition();
    PVector positionStart = g.getStartPosition();
    PVector direction = g.getDirection();
    float speed = g.getSpeed();
    long duration = g.getDuration();
    float durationSeconds = g.getDurationInSeconds();

    //Write DATA to Json on swipe
    swipeCount = true;
    swipe = 1;
    println("Data Saved");
    writeJson();
    swipe = 0;
}
```

Getting pinch gesture from user.

```
void pinch() {  
    float distance = dist(fingerThumbPosition.x, fingerThumbPosition.y,  
fingerIndexPosition.x, fingerIndexPosition.y);  
    //println(distance);  
    if (distance <= 40) {  
        pinchStatus = true;  
        //println("pinch");  
    } else {  
        pinchStatus = false;  
        DraggingMode = -1;  
    }  
}
```



Pinch + Drag .

```
void pinchDragged() {  
    float d1 = dist(fingerThumbPosition.x, fingerThumbPosition.y, x1, y1);  
    if ((d1 <= 40) && pinchStatus == true)  
    {  
        y1 = fingerThumbPosition.y;  
        cVald1 = y1;  
    }  
    float d2 = dist(fingerThumbPosition.x, fingerThumbPosition.y, x2, y2);  
    if ((d2 <= 30) && pinchStatus == true)  
    {  
        y2 = fingerThumbPosition.y;  
        cVald2 = y2;  
    }  
    float d3 = dist(fingerThumbPosition.x, fingerThumbPosition.y,  
    if ((d3 <= 30) && pinchStatus == true)  
    {  
        cx1 = fingerThumbPosition.x;  
        cy1 = fingerThumbPosition.y;  
    }  
    float d4 = dist(fingerThumbPosition.x, fingerThumbPosition.y,  
    if ((d4 <= 30) && pinchStatus == true)  
    {  
        cx2 = fingerThumbPosition.x;  
        cy2 = fingerThumbPosition.y;  
    }  
    if(pinchStatus == true){  
        pWidth = 30;  
    } else {  
        pWidth = 20;  
    }  
}
```



Save data on swipe gesture to JSON file.

```
void writeJson(){
    if(swipe == 1){
        println(swipe);
        JSONObject newPos = new JSONObject();

        //Creating New Json position object
        JSONObject position = new JSONObject();
        position.setFloat("x1", x1);
        position.setFloat("y1", y1);
        position.setFloat("cx1", cx1);
        position.setFloat("cy1", cy1);
        position.setFloat("cx2", cx2);
        position.setFloat("cy2", cy2);
        position.setFloat("x2", x2);
        position.setFloat("y2", y2);

        //Add position to bezs
        newPos.setJSONObject("position", position);

        //Write data to json
        JSONArray bezData = json.getJSONArray("bezs");
        bezData.append(newPos);

        saveJSONObject(json,"dataCopy.json");
        swipeCount = false;
        swipe = 0;
    }
}
```



Deep understanding with a in-action demo video

Click on image or the link below.



<https://vimeo.com/317577462>