

SPIELRAUM PHONE

Technische Documentatie

Kernmodule 2 IxD

Daan Walder

Algemene omschrijving:

De Spielraum Phone is een concept dat is afgeleid van kinderspeelgoed waar ikzelf vroeger veel mee speelde. Het idee is namelijk vrij simpel. Je hebt, net zoals bij oude huistelefoons, twee of meer bedraden telefoons, die allen met elkaar zijn verbonden. Zodra je beide de telefoon opneemt kun je met elkaar praten, zonder dat er iemand daadwerkelijk naar je heeft gebeld.

Technische omschrijving:

Voor het prototype van de SPIELRAUM PHONE heb ik gewerkt in openFrameworks. Ik heb hierin drie verschillende schermen gemaakt, die worden weergegeven afhankelijk van de "fase" waarin de telefoons zich bevindt. Door middel van een LDR-sensor, die ik in twee telefoons heb gebouwd, kan openFrameworks via de Arduino checken of een telefoon is opgepakt. Als de horen van de telefoon namelijk is neergelegd detecteert de Arduino geen licht. Als dit bij beide telefoons het geval is dan laat openFrameworks het eerste scherm zien. Als dan een van de telefoon wordt opgepakt dan wordt het tweede scherm getoond. Als beide telefoons zijn opgepakt, dan wordt het derde scherm getoond.



```

1  #include "ofApp.h"
2
3  #define PIN_LDR 0
4  #define PIN_SECONDLDR 1
5
6  void ofApp::setup() {
7      ofAddListener(arduino.EInitialized, this, &ofApp::setupArduino);
8      arduino.connect("COM3", 57600);
9      arduino.sendFirmwareVersionRequest();
10
11      ofSetFullscreen(0);
12
13      startScreen.load("startScreen.mp4");
14      callingScreen.load("callingScreen.mp4");
15      onPhoneScreen.load("onPhoneScreen.mp4");
16  }
17
18  void ofApp::setupArduino(const int& version) {
19      ofRemoveListener(arduino.EInitialized, this, &ofApp::setupArduino);
20      ofLog() << "Arduino firmware found: " << arduino.getFirmwareName()
21          << " v" << arduino.getMajorFirmwareVersion() << "." << arduino.getMinorFirmwareVersion();
22      arduino.sendAnalogPinReporting(PIN_LDR, ARD_ANALOG);
23      arduino.sendAnalogPinReporting(PIN_SECONDLDR, ARD_ANALOG);
24
25      ofAddListener(arduino.EAnalogPinChanged, this, &ofApp::analogPinChanged);
26  }
27
28  void ofApp::analogPinChanged(const int& pinNum) {
29      // get value with arduino.getAnalog(pinNum);
30      ofLogNotice() << "Analog Pin " << 0 << " value: " << arduino.getAnalog(0) << endl;
31      ofLogNotice() << "Analog Pin " << 1 << " value: " << arduino.getAnalog(1) << endl;
32  }
33
34
35  void ofApp::update() {
36      arduino.update();
37      callingScreen.update();
38      onPhoneScreen.update();
39      startScreen.update();
40  }
41
42  void ofApp::draw() {
43      if (arduino.getAnalog(0) <= 3 && arduino.getAnalog(1) <= 8)
44      {
45          callingScreen.stop();
46          onPhoneScreen.stop();
47          startScreen.play();
48          std::printf("startScreen");
49          startScreen.draw(0, 0, ofGetScreenWidth(), ofGetScreenHeight());
50
51      }
52
53      else if (arduino.getAnalog(0) >= 3 && arduino.getAnalog(1) <= 8)
54      {
55          startScreen.stop();
56          onPhoneScreen.stop();
57          callingScreen.play();
58          std::printf("calling");
59          callingScreen.draw(0, 0, ofGetScreenWidth(), ofGetScreenHeight());
60      }
61
62      else if (arduino.getAnalog(0) <= 3 && arduino.getAnalog(1) >= 8)
63      {
64          startScreen.stop();
65          onPhoneScreen.stop();
66          callingScreen.play();
67          std::printf("calling");
68          callingScreen.draw(0, 0, ofGetScreenWidth(), ofGetScreenHeight());
69      }
70
71      else if (arduino.getAnalog(0) >= 3 && arduino.getAnalog(1) >= 8)
72      {
73          callingScreen.stop();
74          startScreen.stop();
75          onPhoneScreen.play();
76          std::printf("connected");
77          onPhoneScreen.draw(0, 0, ofGetScreenWidth(), ofGetScreenHeight());
78      }
79
80  }

```