

```
1 // Kenneth Ahlmann Moothedath Johansen
2 // Johan Maack Svendsen
3
4
5 Bit[] bits = new Bit[8]; // array with room for 8 on/off Bit instances
6 int decimal = 0;
7 int binary = 0;
8 boolean sign = false; //ny boolean til at bestemme fortegn
9 PFont font;
10
11 void setup() {
12     size(600, 300);
13     noStroke();
14     font = createFont("Arial", 48, true); // Windows 10 have a wrong font path
15     for (int i = 0; i < bits.length; i++) {
16         bits[i] = new Bit(i);
17     } // for
18 } // setup
19
20 void draw() {
21     background(0);
22     for (int i = 0; i < bits.length; i++) {
23         bits[i].display();
24         fill(255);
25         int bitValue = 1 << (bits.length - i - 1); // very fast calculation of 2^i
26         if (bits.length - i - 1 != 7) {
27             text(bitValue, width/9 * bits[i].position - 10, 50);
28         } else {
29             text("fortegn", width/9 * bits[i].position - 10, 50);
30         } // if else for at sørge for den ikke skriver 128 ved mest betydende bit men fortegn i stedet
31
```

```
32     } // for
33     fill(255);
34     textFont(font, 48);
35     textAlign(RIGHT);
36     text(nf(binary, 8), width/9*8, 180);
37     text(decimal, width/9*8, 230);
38     textAlign(LEFT);
39     text("Binært:", width/9, 180);
40     text("Decimalt:", width/9, 230);
41     textFont(font, 18);
42     fill(0, 255, 255); // Cyan text
43     text("Klik en bit for at tænde (og addere værdien  $2^n$ ) eller sluk for en bit.", width/22, 25);
44 } // draw
45
46 void keyReleased() {
47     decimal = 0;
48     binary = 0;
49     for (int i = 0; i < bits.length; i++) {
50         bits[i].updateKey();
51         if (decimal < 0) { // tester om det nuværende decimal er negativt eller positivt
52             decimal -= bits[i].value; // trækker fra hvis det er negativt
53         } else {
54             decimal += bits[i].value; // lægger til hvis det er positivt
55         }
56
57         if (sign == true) {
58             if (decimal > 0){
59                 decimal = decimal * -1;
60             } // tester om sign er true og ændre fortegnet til negativt hvis det nuværende decimal er positivt
61         } else {
62             if (decimal < 0) {
```

```
63     decimal = decimal * -1;
64 }
65 } // ændre fortegnet til positivt hvis det nuværende var negativt
66     binary += bits[i].digit;
67 } // for
68 } // keyReleased
69
70 void mouseReleased() {
71     decimal = 0;
72     binary = 0;
73     for (int i = 0; i < bits.length; i++) {
74         bits[i].updateMouse();
75         if (decimal < 0) { // tester om det nuværende decimal er negativt eller positivt
76             decimal -= bits[i].value; // trækker fra hvis det er negativt
77         } else {
78             decimal += bits[i].value; // lægger til hvis det er positivt
79         }
80
81         if (sign == true) {
82             if (decimal > 0 ){
83                 decimal = decimal * -1;
84             } // tester om sign er true og ændre fortegnet til negativt hvis det nuværende decimal er positivt
85         } else {
86             if (decimal < 0) {
87                 decimal = decimal * -1;
88             }
89         } // ændre fortegnet til positivt hvis det nuværende var negativt
90
91         binary += bits[i].digit;
92     } // for
93 } // mouseReleased
```

```

94
95  class Bit {          // Bit object class
96      int position;
97      color colour = (55); // Grey
98      int value = 0;
99      int digit = 0;
100
101      Bit(int pos) {
102          position = pos + 1;
103      }
104
105      void display() {
106          fill(colour);
107          ellipse(width/9 * position, 80, 50, 50);
108      }
109
110      void updateKey() {
111          if (key == position + 48) {
112              switch(colour) {
113                  case (55):
114                      colour = (255);
115                      digit = int (pow(10, 8 - position));
116                      if (8 - position != 7) { //tester om det ikke er sign-bitten og opløfter 2 i den placeringen af bitten
117                          value = int (pow(2, 8 - position));
118                      } else { // ædnre sign til true hvis det var sign-bitten der var ændret
119                          value = 0;
120                          sign = true;
121                      }
122                      break;
123                  case (255):
124                      colour = (55);

```

```
125     value = 0;
126     digit = 0;
127     if (8 - position == 7) {
128         sign = false;
129     } // tester om det er sign-bitten og ændre sign til false hvis det er den der er slået fra
130     break;
131 } // switch
132 } // if
133 } // updateKey
134
135 void updateMouse() {
136     if (onCircle(width/9 * position, 80, 50)) {
137         switch(colour) {
138             case (55):
139                 colour = (255);
140                 digit = int (pow(10, 8 - position));
141                 if (8 - position != 7) { //tester om det ikke er sign-bitten og opløfter 2 i den placeringen af bitten
142                     value = int (pow(2, 8 - position));
143                 } else { // ædnre sign til true hvis det var sign-bitten der var ændret
144                     value = 0;
145                     sign = true;
146                 }
147                 break;
148             case (255):
149                 colour = (55);
150                 value = 0;
151                 digit = 0;
152                 if (8 - position == 7) {
153                     sign = false;
154                 } // tester om det er sign-bitten og ændre sign til false hvis det er den der er slået fra
155
```

```
156         break;
157     } // switch
158 } // onCircle
159 } // updateMouse
160 } // class
161
162 boolean onCircle(int x, int y, int diameter) {
163     float distX = x - mouseX;
164     float distY = y - mouseY;
165     int radius = diameter / 2;
166     if ( sqrt(sq(distX) + sq(distY)) < radius ) {
167         return true;
168     } else {
169         return false;
170     } // if
171 } // end onCircle
172
173 // end
```