

```
1 using Microsoft.SPOT;
2 using Microsoft.SPOT.Hardware;
3
4 //Nowendig für Visual Studio 2015
5 namespace System.Diagnostics
6 {
7     public enum DebuggerBrowsableState
8     {
9         Never,
10        Collapsed,
11        RootHidden
12    }
13 }
14
15 class Program
16 {
17     private BrainPad.Image imageOben = new BrainPad.Image(30, 30);
18     private BrainPad.Image imageElement = new BrainPad.Image(32, 32);
19     private BrainPad.Image imageLinks = new BrainPad.Image(30, 30);
20     private BrainPad.Image imageRechts = new BrainPad.Image(30, 30);
21     private BrainPad.Image imageGrau=new BrainPad.Image(28,28);
22     private BrainPad.Image imageRot = new BrainPad.Image(28, 28);
23     private BrainPad.Image imageGelb = new BrainPad.Image(28, 28);
24     private BrainPad.Image imageGruen = new BrainPad.Image(28, 28);
25
26     private int modus;
27     private bool stop;
28
29     public void BrainPadSetup()
30     {
31         BrainPad.Button.ButtonPressed += Button_ButtonPressed;
32
33         getImage(imageOben, BrainPad\_Projekt\_07.Properties.Resources.BinaryResources.Oben);
34         getImage(imageElement, BrainPad\_Projekt\_07.Properties.Resources.BinaryResources.Element);
35         getImage(imageLinks, BrainPad\_Projekt\_07.Properties.Resources.BinaryResources.Links);
36         getImage(imageRechts, BrainPad\_Projekt\_07.Properties.Resources.BinaryResources.Rechts);
37         getImage(imageGrau, BrainPad\_Projekt\_07.Properties.Resources.BinaryResources.Grau);
38         getImage(imageRot, BrainPad\_Projekt\_07.Properties.Resources.BinaryResources.Rot);
39         getImage(imageGelb, BrainPad\_Projekt\_07.Properties.Resources.BinaryResources.Gelb);
40         getImage(imageGruen, BrainPad\_Projekt\_07.Properties.Resources.BinaryResources.Gruen);
41
42         BrainPad.Display.DrawImage(65, 2, imageOben);
43         BrainPad.Display.DrawImage(64, 12, imageElement);
44         BrainPad.Display.DrawImage(64, 44, imageElement);
45         BrainPad.Display.DrawImage(64, 76, imageElement);
46         BrainPad.Display.DrawImage(32, 13, imageLinks);
47         BrainPad.Display.DrawImage(32, 46, imageLinks);
48         BrainPad.Display.DrawImage(32, 78, imageLinks);
```

```
49     BrainPad.Display.DrawImage(98, 13, imageRechts);
50     BrainPad.Display.DrawImage(98, 46, imageRechts);
51     BrainPad.Display.DrawImage(98, 78, imageRechts);
52
53     anzeigen(false, false, false);
54     modus = 0;
55     stop = false;
56 }
57
58 public void BrainPadLoop()
59 {
60     steuern();
61 }
62
63 private void Button_ButtonPressed(BrainPad.Button.DPad button,           ↗
    BrainPad.Button.State state)
64 {
65     if (button == BrainPad.Button.DPad.Right)
66     {
67         if (state == BrainPad.Button.State.Pressed)
68         {
69             if (modus < 3)
70                 modus++;
71             else
72                 modus = 0;
73         }
74     }
75     if (button == BrainPad.Button.DPad.Left)
76     {
77         if (modus == 3)
78         {
79             stop = true;
80         }
81     }
82 }
83
84 private void getImage(BrainPad.Image image,                               ↗
    BrainPad_Projekt_07.Properties.Resources.BinaryResources resource)
85 {
86     int info;
87     byte[] data;
88     BrainPad.Color color = new BrainPad.Color();
89
90     data = (byte[])ResourceUtility.GetObject                               ↗
        (BrainPad_Projekt_07.Properties.Resources.ResourceManager, resource);
91     info = data[0xA];
92     for (int Y = image.Height - 1; Y >= 0; Y--)
93     {
94         for (int X = 0; X < image.Width; X++)
95         {
96             byte h = data[info++];
97             byte n = data[info++];
98             color.B = (byte)(h & 0x1f);
99             color.G = (byte)(((h & 0xe0) >> 5) + ((n & 0x07) << 3));
100            color.R = (byte)((n & 0xf8) >> 3);
101            image.SetPixel(Y, X, color);
```

```
102     }
103     }
104 }
105
106 private void anzeigen(bool Rot, bool Gelb, bool Gruen)
107 {
108     BrainPad.TrafficLight.TurnOffAllLights();
109     if (Rot)
110     {
111         BrainPad.Display.DrawImage(66, 14, imageRot);
112         BrainPad.TrafficLight.TurnRedLightOn();
113     }
114     else
115         BrainPad.Display.DrawImage(66, 14, imageGrau);
116     if (Gelb)
117     {
118         BrainPad.Display.DrawImage(66, 46, imageGelb);
119         BrainPad.TrafficLight.TurnYellowLightOn();
120     }
121     else
122         BrainPad.Display.DrawImage(66, 46, imageGrau);
123     if (Gruen)
124     {
125         BrainPad.Display.DrawImage(66, 78, imageGruen);
126         BrainPad.TrafficLight.TurnGreenLightOn();
127     }
128     else
129         BrainPad.Display.DrawImage(66, 78, imageGrau);
130 }
131
132 private void steuern()
133 {
134     switch (modus)
135     {
136     case 0:
137         BrainPad.Display.DrawText(57, 115, "Modus: 0",
138             BrainPad.Color.White);
139         anzeigen(false, false, false);
140         break;
141     case 1:
142         BrainPad.Display.DrawText(57, 115, "Modus: 1",
143             BrainPad.Color.White);
144         anzeigen(false, true, false);
145         BrainPad.Wait.Seconds(1.0);
146         anzeigen(false, false, false);
147         BrainPad.Wait.Seconds(1.0);
148         break;
149     case 2:
150         BrainPad.Display.DrawText(57, 115, "Modus: 2",
151             BrainPad.Color.White);
152         anzeigen(true, false, false);
153         BrainPad.Wait.Seconds(2.0);
154         anzeigen(true, true, false);
155         BrainPad.Wait.Seconds(2.0);
156         anzeigen(false, false, true);
157         BrainPad.Wait.Seconds(2.0);
```

```
155     anzeigen(false, true, false);
156     BrainPad.Wait.Seconds(2.0);
157     break;
158     case 3:
159         BrainPad.Display.DrawText(57, 115, "Modus: 3",
160             BrainPad.Color.White);
161         if (!stop)
162         {
163             anzeigen(true, false, false);
164         }
165         else
166         {
167             anzeigen(true, true, false);
168             BrainPad.Wait.Seconds(1.0);
169             anzeigen(false, false, true);
170             BrainPad.Wait.Seconds(1.0);
171             anzeigen(false, true, false);
172             BrainPad.Wait.Seconds(1.0);
173             stop = false;
174         }
175     }
176 }
177 }
178
179
```