

```

1 //Breite und Höhe
2 int b;
3 int h;
4 //Spieler links
5 int x;
6 int y;
7 //Spieler rechts
8 int x2;
9 int y2;
10 //Tasten
11 boolean kTaste;
12 boolean iTaste;
13 boolean wTaste;
14 boolean sTaste;
15 //Ball
16 int xB;
17 int yB;
18 int r;
19 int vx;
20 int vy;
21
22 //spielfeld
23 int spielBreite;
24 int spielHoehe;
25
26 boolean links1, rechts1, oben1, unten1;
27 boolean links2, rechts2, oben2, unten2;
28
29 int xEdge1;
30 int yEdge1;
31
32 int xEdge2;
33 int yEdge2;
34
35 int punkteSpieler1;
36 int punkteSpieler2;
37
38 public void setup()
39 {
40     size(400, 400);
41     spielBreite = 400;
42     spielHoehe = 400;
43
44     xB = 200;
45     yB=200;
46
47     h=40;
48     b=10;
49
50     x=40;
51     y=100;
52
53     x2=spielBreite-40-b;
54     y2=100;
55
56     kTaste=false;
57     iTaste=false;
58     wTaste=false;
59     sTaste=false;
60
61     vx=2;
62     vy=2;

```

```

63
64     r = 10;
65
66     xEdge1=0;
67     yEdge1=0;
68     xEdge2=0;
69     yEdge2=0;
70
71     punkteSpieler1=0;
72     punkteSpieler2=0;
73 }
74
75 void draw()
76 {
77     background(0, 0, 255);
78
79     textSize(20);
80     text(punkteSpieler1, spielBreite/2 -20, 50);
81     text(punkteSpieler2, spielBreite/2 + 20, 50);
82
83     rect(x, y, b, h);
84     rect(x2, y2, b, h);
85
86     if (kTaste&&y2<spielHoehe-h)
87     {
88         y2=y2+5;
89     }
90     if (iTaste&&y2>0)
91     {
92         y2=y2-5;
93     }
94
95     if (wTaste&&y>0)
96     {
97         y=y-5;
98     }
99     if (sTaste&&y<spielHoehe-h)
100    {
101        y=y+5;
102    }
103
104
105    ellipse(xB, yB, 2*r, 2*r);
106    xB = xB+vx;
107    yB=yB+vy;
108
109
110    //Kollisionsabfrage
111
112    //Finde Ecke zur Kollision mit Spieler 1
113    if (xB<x) {
114        xEdge1 = x;
115        links1 = true;
116        rechts1 = false;
117    } else
118    {
119        if (xB<x+b)
120        {
121            xEdge1 = xB;
122            rechts1 = false;
123            links1 = false;
124        } else {
125            xEdge1 = x+b;

```

```

126     links1 = false;
127     rechts1 = true;
128 }
129 }
130
131 if (yB<y) {
132     yEdge1 = y;
133     oben1 = true;
134     unten1 = false;
135 } else
136 {
137     if (yB<y+h)
138     {
139         yEdge1=yB;
140         unten1 = false;
141         oben1 = false;
142     } else {
143         yEdge1 = y+h;
144         oben1 = false;
145         unten1 = true;
146     }
147 }
148
149 //Finde Ecke zur Kollision mit Spieler 2
150 if (xB<x2) {
151     xEdge2 = x2;
152     links2 = true;
153     rechts2 = false;
154 } else
155 {
156     if (xB<x2+b)
157     {
158         xEdge2 = xB;
159         rechts2 = false;
160         links2 = false;
161     } else {
162         xEdge2 = x2+b;
163         links2 = false;
164         rechts2 = true;
165     }
166 }
167
168 if (yB<y2) {
169     yEdge2 = y2;
170     oben2 = true;
171     unten2 = false;
172 } else
173 {
174     if (yB<y2+h)
175     {
176         yEdge2=yB;
177         unten2 = false;
178         oben2 = false;
179     } else {
180         yEdge2 = y2+h;
181         oben2 = false;
182         unten2 = true;
183     }
184 }
185
186 //Berechne ob Kollision mit Spieler 1
187 if (bestimmeAbstand(xEdge1, yEdge1, xB, yB)<r)

```

```

188 {
189     if ((oben1 &&links1&& vy>0))
190     {
191         vy=vy*(-1);
192     }
193     if ((oben1 &&rechts1&& vy>0))
194     {
195         vy=vy*(-1);
196     }
197     if ((unten1 &&links1&& vy<0))
198     {
199         vy=vy*(-1);
200     }
201     if ((unten1 &&rechts1&& vy<0))
202     {
203         vy=vy*(-1);
204     }
205
206     if ((links1 &&oben1&& vx>0))
207     {
208         vx=vx*(-1);
209     }
210     if ((links1 &&unten1&& vx>0))
211     {
212         vx=vx*(-1);
213     }
214     if ((rechts1 &&oben1&& vx<0))
215     {
216         vx=vx*(-1);
217     }
218     if ((rechts1 &&unten1&& vx<0))
219     {
220         vx=vx*(-1);
221     }
222     if (rechts1&&(!oben1&&!unten1)||links1&&(!oben1&&!unten1))
223     {
224         vx=vx*(-1);
225     }
226     if (oben1&&(!rechts1&&!links1)||unten1&&(!rechts1&&!links1))
227     {
228         vy=vy*(-1);
229     }
230 }
231 //Berechne ob Kollision mit Spieler 2
232 else if (bestimmeAbstand(xEdge2, yEdge2, xB, yB)<r)
233 {
234     if ((oben2 &&links2&& vy>0))
235     {
236         vy=vy*(-1);
237     }
238     if ((oben2 &&rechts2&& vy>0))
239     {
240         vy=vy*(-1);
241     }
242     if ((unten2 &&links2&& vy<0))
243     {
244         vy=vy*(-1);
245     }
246     if ((unten2 &&rechts2&& vy<0))
247     {
248         vy=vy*(-1);
249     }

```

```

250
251     if ((links2 &&oben2&& vx>0))
252     {
253         vx=vx*(-1);
254     }
255     if ((links2 &&unten2&& vx>0))
256     {
257         vx=vx*(-1);
258     }
259     if ((rechts2 &&oben2&& vx<0))
260     {
261         vx=vx*(-1);
262     }
263     if ((rechts2 &&unten2&& vx<0))
264     {
265         vx=vx*(-1);
266     }
267     if (rechts2&&(!oben2&&!unten2)||links2&&(!oben2&&!unten2))
268     {
269         vx=vx*(-1);
270     }
271     if (oben2&&(!rechts2&&!links2)||unten2&&(!rechts2&&!links2))
272     {
273         vy=vy*(-1);
274     }
275 }
276 //Sonst Kollision mit Rand des Spielfeldes?
277 else {
278     if (xB-r<0)
279     {
280         xB=200;
281         punkteSpieler2=punkteSpieler2+1;
282         vy=2;
283         vx=2;
284     }
285     if (xB+r>spielBreite)
286     {
287         xB=200;
288         punkteSpieler1=punkteSpieler1+1;
289         vy=2;
290         vx=2;
291     }
292
293     //Kollision mit Rand
294     if (yB-r<0)
295     {
296         vy=vy*(-1);
297         yB=r;
298     }
299     if (yB+r>spielHoehe)
300     {
301         vy=vy*(-1);
302         yB=spielHoehe-r;
303     }
304 }
305 }
306
307 //Hilfsmethode für Kollision
308 double bestimmeAbstand(int x1, int y1, int x2, int y2)
309 {
310     return sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
311 }

```

```
312
313 void keyReleased()
314 {
315
316     if (key=='s')
317     {
318         sTaste = false;
319     }
320     if (key=='w')
321     {
322         wTaste = false;
323     }
324     if (key=='k')
325     {
326         kTaste = false;
327     }
328     if (key=='i')
329     {
330         iTaste = false;
331     }
332 }
333 void keyPressed()
334 {
335     if (key=='s')
336     {
337         sTaste = true;
338     }
339     if (key=='w')
340     {
341         wTaste = true;
342     }
343     if (key=='k')
344     {
345         kTaste = true;
346     }
347     if (key=='i')
348     {
349         iTaste = true;
350     }
351 }
352
353
354
355
356
```