

```

1 %scout bats1, bats2;
2 window bats1pos;
3 point bats1pos;
4 integer width=50;
5 string bats1rec="";
6 string bats2rec="";
7 bats1pos = (25,50);
8 bats1 = {
9     frame : ([bats1pos, 1, width]),
10    string : bats1rec,
11    border : 1
12 };
13 bats2 = {
14     frame : ([bats1pos+(0.c,2.r), 1, width]),
15    string : bats2rec,
16    border : 1
17 };
18
19
20
21 %geden
22 bats1runs = []; /* list of runs of batsman 1 */
23 bats2runs = []; /* list of runs of batsman 2 */
24
25
26 bats1rec is mkstr(bats1runs); /* a string record of runs by batsman 1 */
27 bats2rec is mkstr(bats2runs); /* a string record of runs by batsman 2 */
28
29 proc updatebats1 : runs
30 {
31     if (batsman == 1) bats1runs = bats1runs // [runs]; else
32     if (batsman == 2) bats2runs = bats2runs // [runs];
33 }
34
35 batsman is ((total % 2)==0) ? 1 : 2 ;
36
37 func mkstr
38 {
39     para l;
40     auto strl;
41     strl = "";
42     while (l!=[]) {
43         strl = strl // str(1[l]) // " ", shift l;
44     }
45     return strl;
46 }
47
48
49 func listsum
50 {
51     para l;
52     auto sum;
53     sum = 0;
54     while (l!=[]) {
55         sum += 1[l], shift l;
56     }
57     return sum;
58 }
59
60 bats1score is listsum(bats1runs);
61 bats2score is listsum(bats2runs);
62
63 total is bats1score + bats2score;
64
65 %scout
66 window teamscore1, teamscore2;
67 window teamtotal;
68 string totalrec="";
69 string recbats1score, recbats2score;
70 teamscore1 = {
71     frame : ([bats1pos+(width+2).c, 0.r], 1, 4)),

```

```

72    string : recbats1score,
73    border : 1
74 );
75 teamscore2 = {
76     frame : ([bats1pos+(width+2).c, 2.r], 1, 4)),
77    string : recbats2score,
78    border : 1
79 };
80 teamtotal = {
81     frame : ([bats1pos+(width+2).c, 4.r], 1, 4)),
82    string : totalrec,
83    border : 1
84 };
85
86 %geden
87 totalrec is str(teamtotal);
88 recbats1score is str(bats1score);
89 recbats2score is str(bats2score);
90
91 %scout
92 screen = <bats1 / bats2 / teamscore1 / teamscore2 / teamtotal>;
93

```