

```

1: ProjectZ.Forest methods for 'arranging'
2:
3: arrange
4:
5:     self arrange: nil
6:
7: arrange: aModel
8:
9:     | fontHeight yValue aPoint |
10:    fontHeight := Node textAttributes lineGrid.
11:    yValue := fontHeight + (Margin y * 2) + Interval y.
12:    nodes
13:        with: (0 to: nodes size - 1)
14:        do:
15:            [:aNode :index |
16:                aNode
17:                    setStatus: UnVisited;
18:                    setLocation: 0 @ (yValue * index)].
19:    aPoint := Point zero.
20:    self rootNodes
21:        do:
22:            [:aNode |
23:                aPoint := self arrange: aNode origin: aPoint model: aModel.
24:                aPoint := 0 @ (aPoint y + Interval y)].
25:    self flushBounds
26:
27: arrange: aNode origin: aPoint model: aModel
28:
29:     | extent subNodes width height x y top h |
30:    aNode setStatus: Visited.
31:    aNode setLocation: aPoint.
32:    self propagate: aModel.
33:    extent := aNode getExtent.
34:    subNodes := self subNodes: aNode.
35:    subNodes isEmpty
36:        ifTrue:
37:            [width := aPoint x + extent x.
38:             height := aPoint y + extent y.
39:             extent := width @ height.
40:             ^extent].
41:    width := aPoint x + extent x.
42:    height := aPoint y.
43:    x := width + Interval x.
44:    y := top := height.
45:    subNodes
46:        do:
47:            [:subNode |
48:                subNode getStatus == UnVisited
49:                    ifTrue:
50:                        [extent := self
51:                            arrange: subNode
52:                                origin: x @ y
53:                                model: aModel.
54:                            h := y + subNode getExtent y.
55:                            y := extent y max: h.

```

```

56:                            width := extent x max: width.
57:                            height := extent y max: height.
58:                            y := y + Interval y]].
59:    y := y - Interval y.
60:    h := aNode getExtent y.
61:    y > (aPoint y + h)
62:        ifTrue:
63:            [y := top + ((y - top - h) / 2).
64:             aNode setLocation: aPoint x @ y.
65:             self propagate: aModel].
66:    height := height max: h.
67:    extent := width @ height.
68:    ^extent
69:
70:
71: ProjectZ.Forest methods for 'flushing'
72:
73: flushBounds
74:
75:     bounds := nil
76:
77:
78: ProjectZ.Forest methods for 'private'
79:
80: propagate: aModel
81:
82:     aModel
83:         ifNotNil:
84:             [(Delay forMilliseconds: SleepTick) wait.
85:              self flushBounds.
86:              aModel changed]

```