Ecopoetics Groundwork

Ira Livingston

v0.6 Mar 2022

0.1 proshap.py by Manuel Gutierrez Algaba

proshap.py (ver 1.1) is a python script written by Manuel Gutierrez Algaba to produce shape definitions from rough 'ascii art'. There is no instruction manual, so here are Donald Arseneau's observations. There is not much of a user interface; look in proshap.py (which is a plain text file) and see how the various 'test' shapes are defined (note the triple-double quotes). Choose one of them, or add a new one, then change the line 'test = test3' to select the desired shape. Execute 'python proshap.py' which will output a definition of "9.40b9.4 0t7.66.8 0.8t5.211.6 1.6t4.413.2 2.4t4.014.0 3.2t3.64.933333st8.5333334.933333st13.466664.933333 4.0t3.62.4t9.62.8t16.42.4 6.4t3.62.0t10.02.0t16.82.0 8.8t3.62.4t9.61.4st11.01.4t16.42.0 11.2t3.63.4jt7.03.4t11.63.4jt15.03.4 12.8t4.06.0t12.06.0 14.4t4.45.2t12.45.2 16.0t4.86.0jt10.86.0 16.8t5.25.6st10.85.6 17.6t5.60.8t14.81.2 19.2t6.02.4t12.03.6 20.8t6.44.4jt10.84.4 22.4e9.4" to the screen and to the file 'result.tex'. The goulish face you see here is the test3 shape. You should be aware that the characters in the ascii input are treated as square, even though they are taller than they are wide, so the output shape specification will be taller and thinner than the input text. There also seems to be a problem with all 'bottoms': flat bottoms of text blocks and of holes are expanded downwards to end at a point. Compare this face to the original face in proshap.py. Warning: These instructions and observations are probably wrong; the author does not program in python so can't even read the code properly. For now, look for proshap.py bundled with shapepar.sty.

0.2 proshap.py by Manuel Gutierrez Algaba

aarati (ver 1.1) is a python script written by Manuel Gutierrez Algaba to produce shape definitions from rough 'ascii art'. There is no instruction manual, so here are Donald Arseneau's observations. There is not much of a user interface; look in proshap.py (which is a plain text file) and see how the various 'test' shapes are defined (note the triple-double quotes). Choose one of them, or add a new one, then change the line 'test = test3' to select the desired picture. Execute 'python proshap.py' which will output a definition of "9.40b9.4
0t5.26.8
4.0t5.26.8
6.4t5.26.8
12.8t5.26.8
14.4t5.26.8
16.8t5.25.6st10.85.6
17.6t5.20.8t14.81.2
19.2t5.22.4t12.03.6
22.4t5.26.8
22.4e9.4" to the screen and to the file 'result.tex'. The goulish face you see here is the test3 shape. You should be aware that the characters in the ascii input are treated as square, even though they are taller than they are wide, so the output shape specification will be taller and thinner than the input text. There also seems to be a problem with all 'bottoms': flat bottoms of text blocks and of holes are expanded downwards to end at a point. Compare this face to the original face in proshap.py. Warning: These instructions and observations are probably wrong; the author does not program in python so can't even read the code properly. For now, look for proshap.py bundled with shapepar.sty.

aarati (ver 1.1) is a python script written by Manuel Gutierrez Algaba to produce shape definitions from rough 'ascii art'. There is no instruction manual, so here are Donald Arseneau's observations. There is not much of a user interface; look in proshap.py (which is a plain text file) and see how the various 'test' shapes are defined (note the triple-double quotes). Choose one of them, or add a new one, then change the line 'test = test3' to select the desired picture. Execute 'python proshap.py' which will output a definition of "9.40b9.4
0t5.26.8
4.0t5.26.8
6.4t5.26.8
12.8t5.26.8
14.4t5.26.8
16.8t7.25.6st10.88.6
17.6t6.20.8t14.88.2
19.2t7.22.4t12.07.6
22.4t7.29.8
22.4e9.4" to the screen and to the file 'result.tex'. The goulish face you see here is the test3 shape. You should be aware that the characters in the ascii input are treated as square, even though they are taller than they are wide, so the output shape specification will be taller an thinner than the input text. There also seems to be a pro with all 'bottoms': flat bo and of holes are expanded do point. Compare this face to the proshap.py. Warning: These instructions and observations are probably wrong; the author does not program in python so can't even read the code properly. For now, look for proshap.py bundled with shapepar.sty.