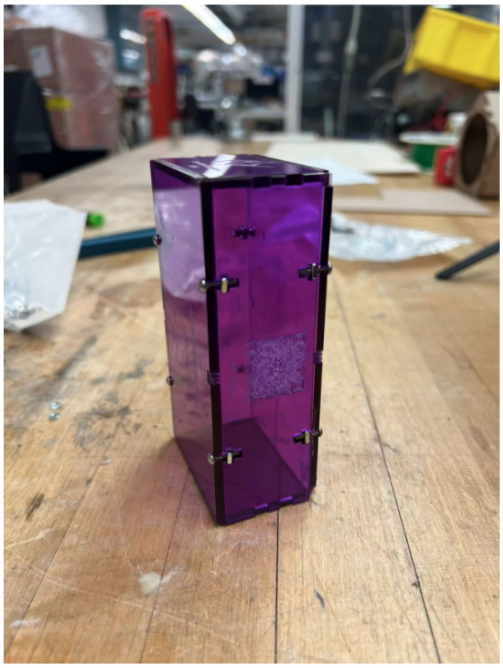
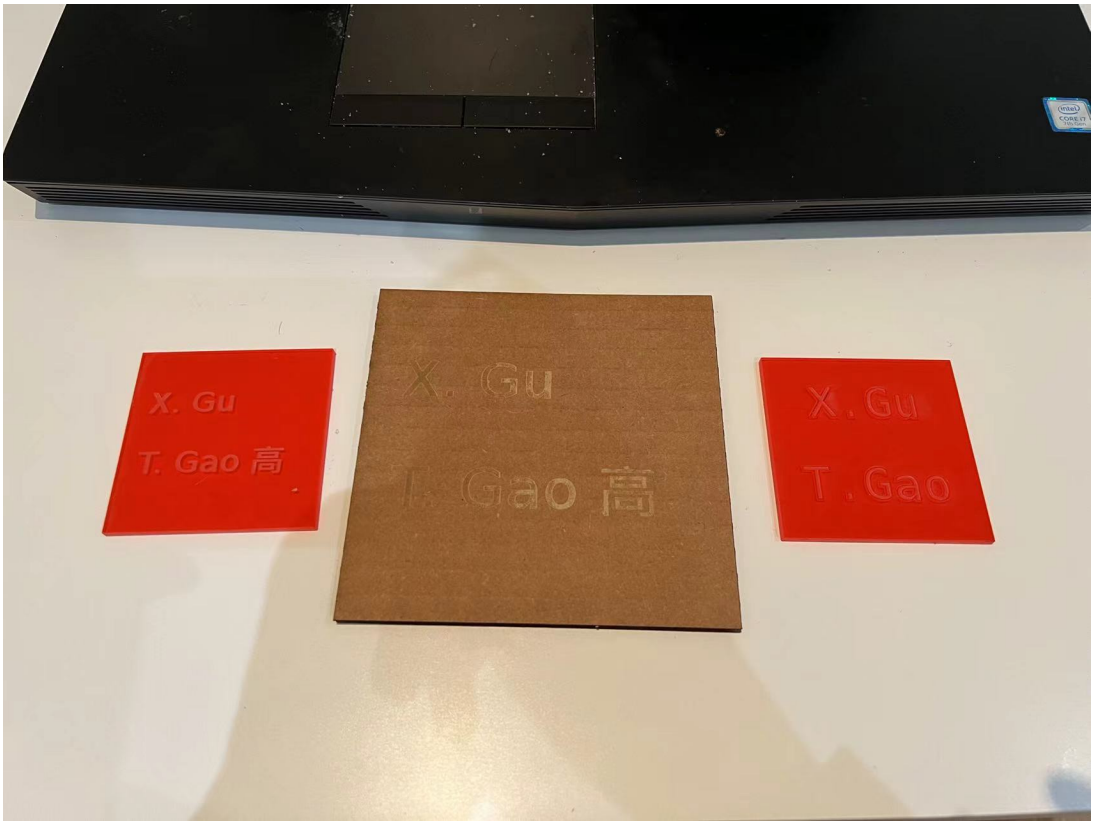
MECE E4606 Digital Manufacturing Assignment 1 Laser Cut Box

Xinsheng Gu (xg2381)
Tianyu Gao (tg2776)
Date/Time submitted: 02/13/2022
Grace hour before submission: 192
Grace hour gained/Used: 0
Grace hour after submission: 192



SMALL RECTANGLE CUT CORRECTLY (SVG+PHOTO)



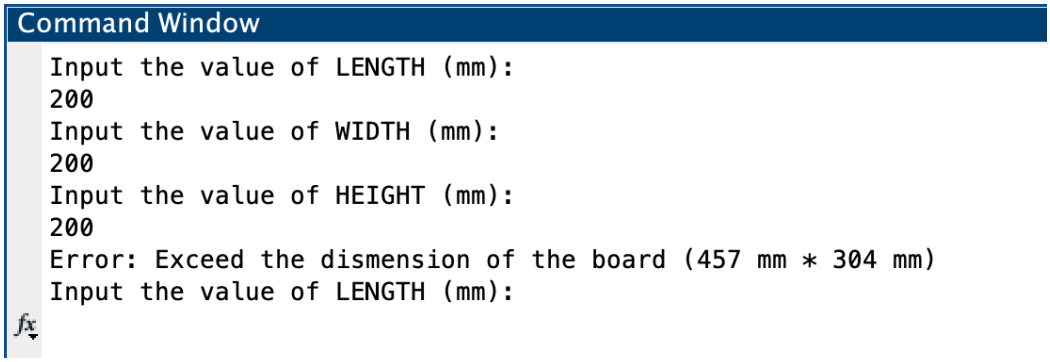

```
<?xml version = "1.0" encoding = "UTF-16" ?>
<svg xmlns = "http://www.w3.org/2000/svg" version = "1.1">
<rect x = "0" y = "0" width = "200" height = "200" fill = "white" stroke-
width = "4" stroke = "black"/>
<!-- <text x = "20" y = "70" font-size="4em">X. Gu</text>
<text x = "20" y = "160" font-size="4em">T. Gao</text> -->
<text x = "15" y = "70" font-size="4em" font-family="monospace" fill="white"
stroke-width="1" stroke="blue">X.Gu</text>
<text x = "15" y = "160" font-size="4em" font-family="monospace" fill="white"
stroke-width="1" stroke="blue">T.Gao</text>
</svg>
```

Software description:

Our software is written in MATLAB and it contains two parts. The main part (a.k.a. autocode.m, see the Appendix) is a program which outputs the svg text file of a flat pattern of a box, while the other part (a.k.a. box.m) is a function which asks users to input the value of the box dimension via keyboard and then the function outputs the parameters for the main program to draw polylines or other patterns. This software aims to generate the flat patterns of boxes automatically and the users are able to see the effect picture immediately.

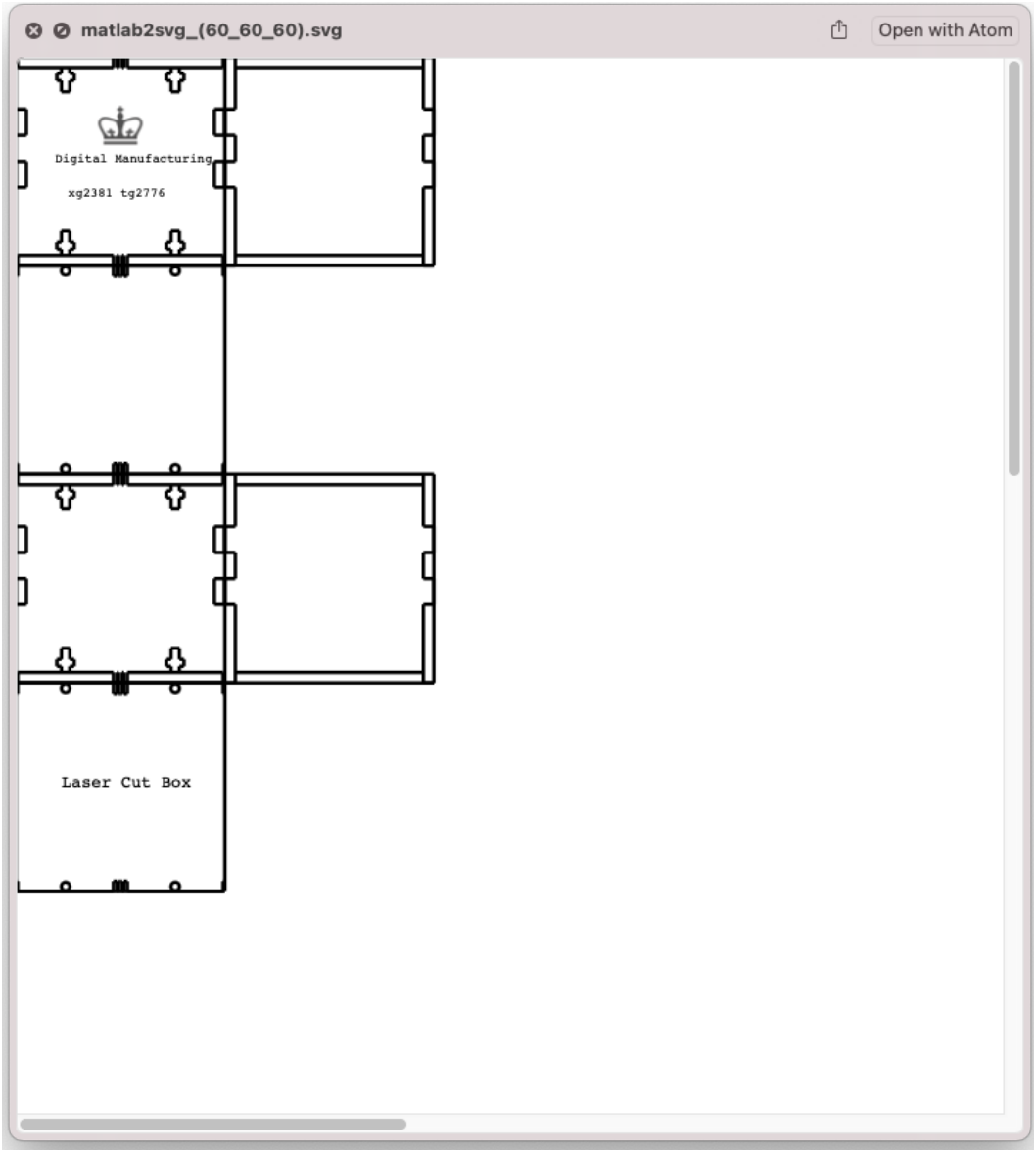
For the autocode.m program, after recalling the parameters from the box.m function, it starts the diary function so that it would automatically turn the output in the command window into a user-specified file, which should be an svg file here. After that, the fprintf and disp function combine the svg language with the box parameters, texts, images and so forth.

For the box.m function. It first requires the user to input the dimensions of a box (length, width and height). If the dimensions are capable for the course-distributed acrylic board to be laser cut, the function will continue to calculate every line and point for the box. Otherwise, the function would detect the error and keep asking the user to retype the dimension again until the dimensions are appropriate. The calculation part contains two parts. The first one is to draw the contour of the six surfaces of a cuboid box and the other is to draw the mortise and tenon structure. Since a cuboid box has symmetrical traits, the function includes some sub-functions to draw a specified pattern and then the function can reuse it.



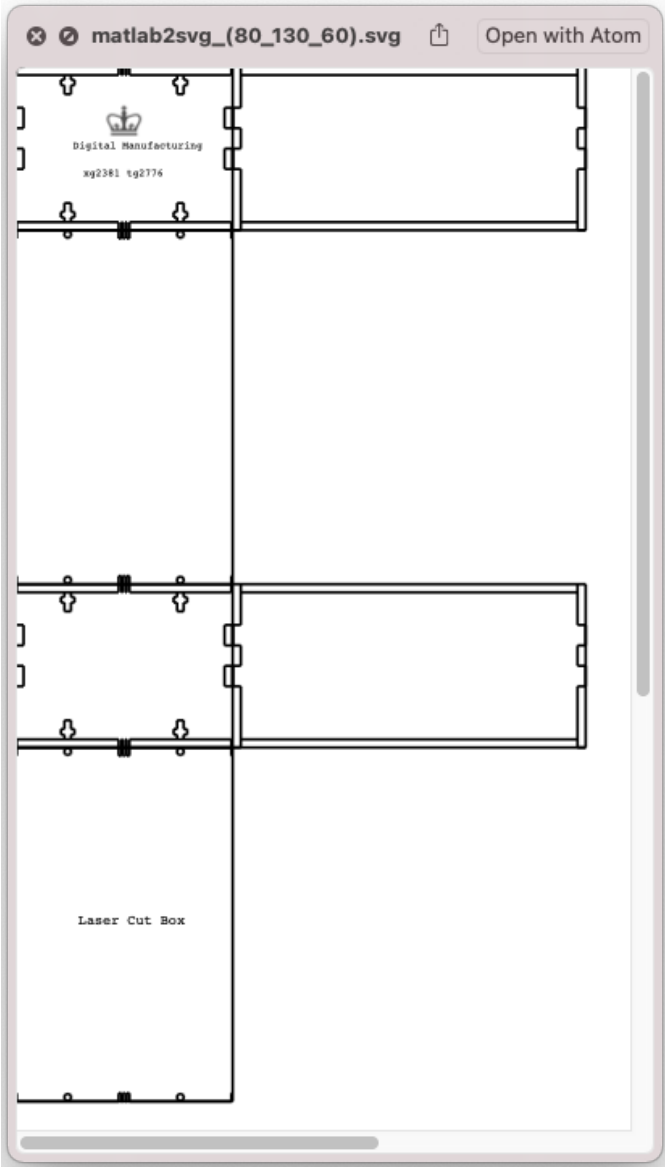
Flat Pattern Example 1:

```
Command Window
  Input the value of LENGTH (mm):
  60
  Input the value of WIDTH (mm):
  60
  Input the value of HEIGHT (mm):
fx 60
```



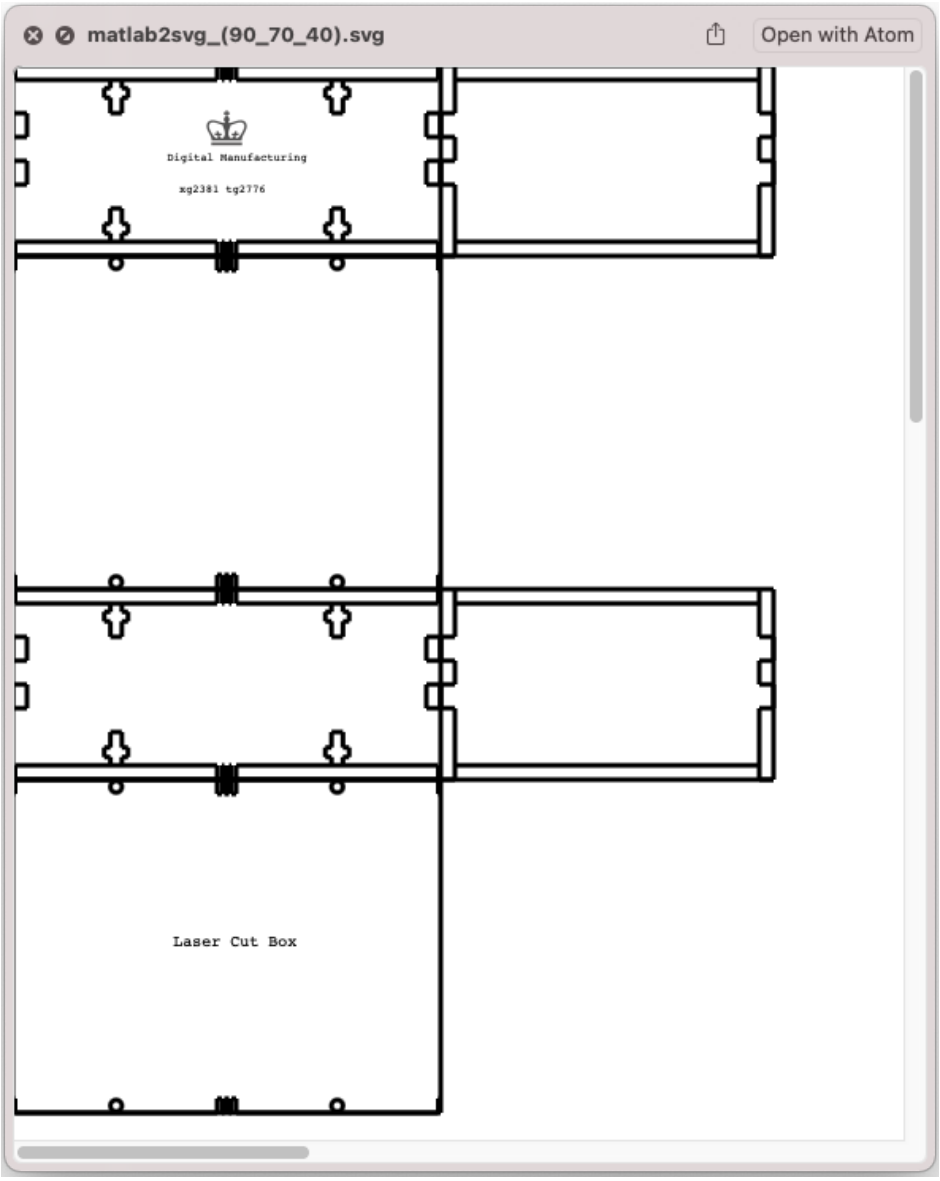
Flat Pattern Example 2:

```
Command Window
  Input the value of LENGTH (mm):
  80
  Input the value of WIDTH (mm):
  130
  Input the value of HEIGHT (mm):
fx 60
```



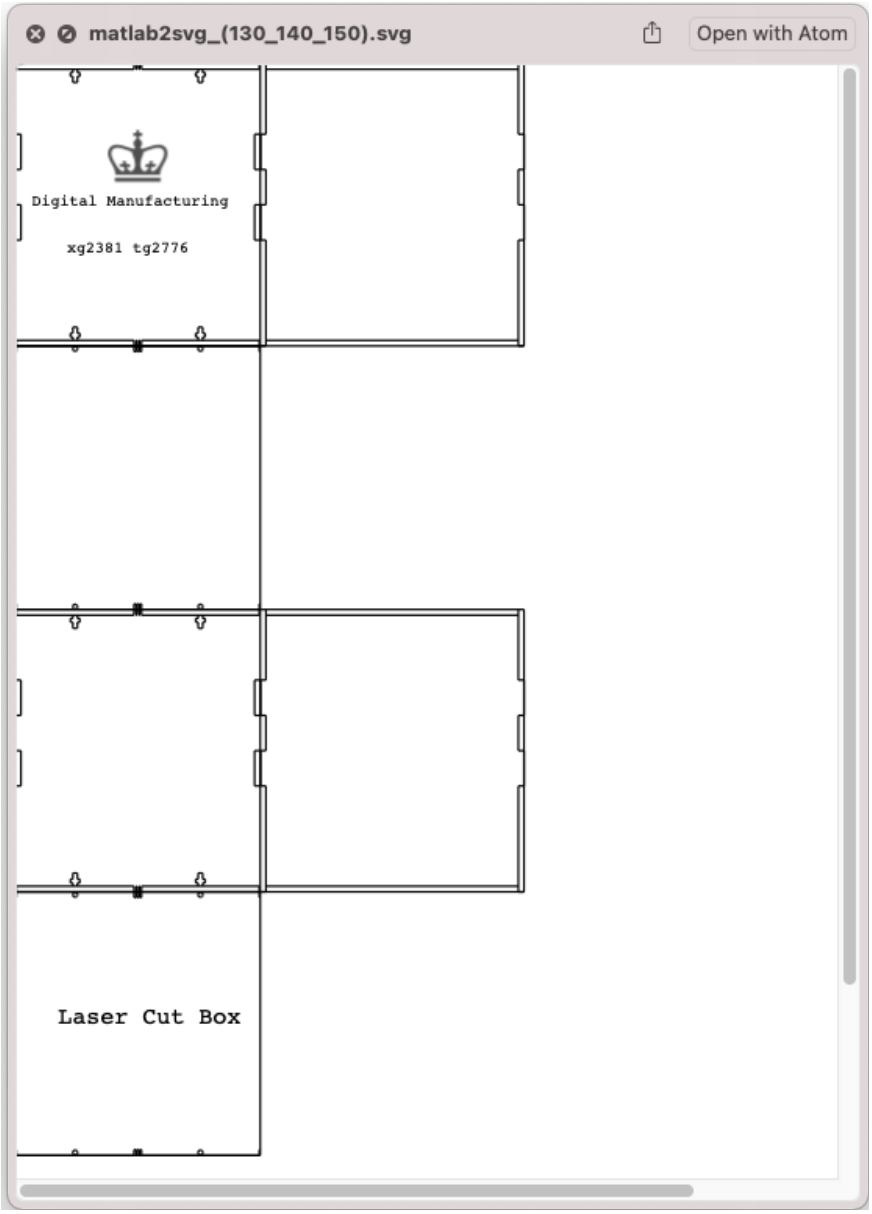
Flat Pattern Example 3:

```
Command Window
  Input the value of LENGTH (mm):
  90
  Input the value of WIDTH (mm):
  70
  Input the value of HEIGHT (mm):
fx 40
```

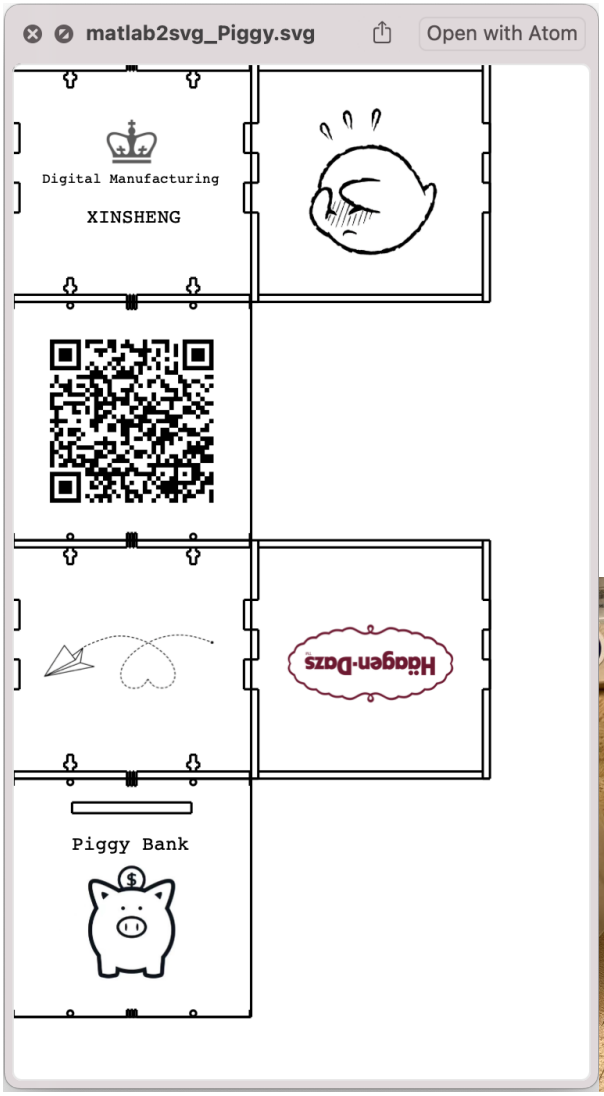


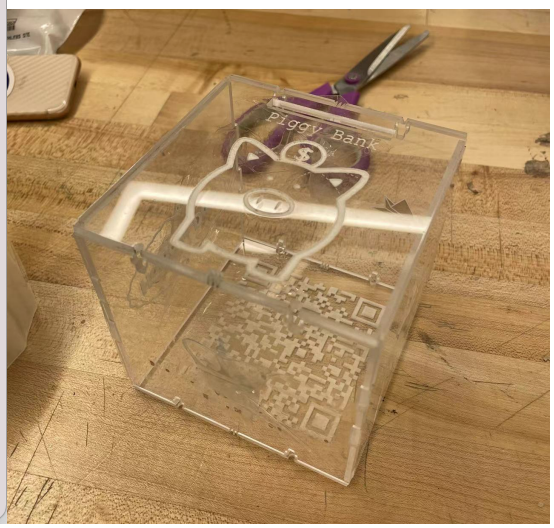
Flat Pattern Example 4:

```
Command Window
  Input the value of LENGTH (mm):
  130
  Input the value of WIDTH (mm):
  140
  Input the value of HEIGHT (mm):
  150
```

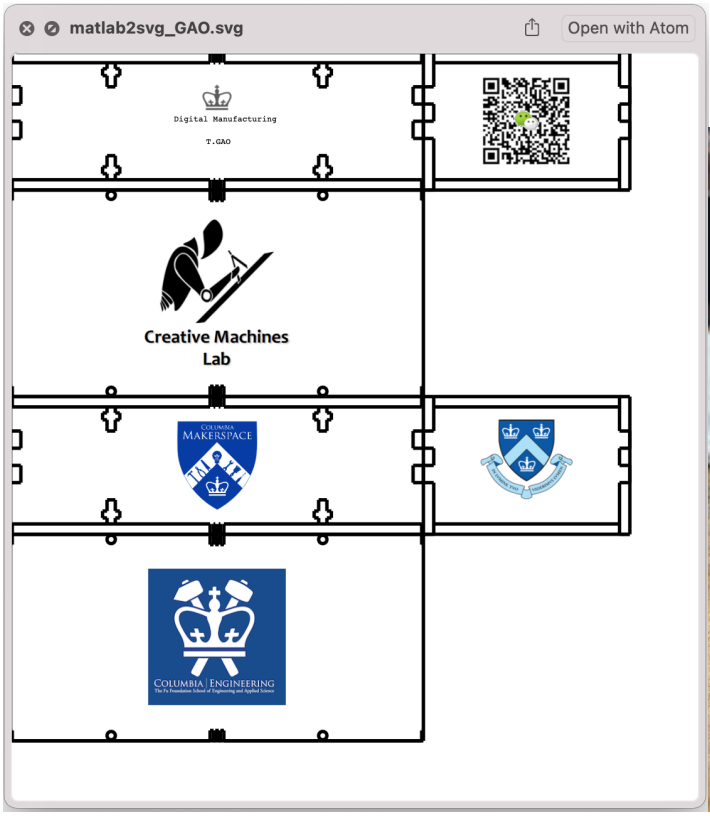


PRINTOUT1:


matlab2svg_Piggy.svg
Open with Atom
Digital Manufacturing
XINSHENG
Häagen-Dazs
Piggy Bank


Piggy Bank

PRINTOUT2:


matlab2svg_GAO.svg
Open with Atom
Digital Manufacturing
T.GAO
Creative Machines
Lab
COLUMBIA MAKERSPACE
COLUMBIA | ENGINEERING

Box image posted on Ed:

Edited Video of the entire process:
(Pending)

References and bibliography:

1. Lipson.H (2.2022). *Laser Cut Box Assignment* [PowerPoint Slides].

APPENDIX:

```
clc;close all;clear;
[L,W,H,P,Cut1_p1,Cut1_p3,Cut2_p8,Cut2_p10,Pattern1_p1,Pattern1_p3,Pattern2_p2,Pattern2_p4,Cir_pos
_p2,Cir_pos_p4,Sc_p1,Sc_p3] = box;
r = 1.2;
dfile ='matlab2svg.svg';
if exist(dfile, 'file')
   delete(dfile);
end
diary on
disp('<?xml version = "1.0" encoding = "UTF-16" ?>')
disp('<svg xmlns = "http://www.w3.org/2000/svg" version = "1.1">')
% fprintf('<polyline points="%d,%d %d,%d" fill="none"
stroke="black"/>\n',p1(1),p1(2),p2(1),p2(2))
seq_P = [1,5;5,6;6,10;11,12;13,14;1,11;2,12;3,13;4,14];
for i = 1:size(seq_P,1)
   disp(['<polyline points="' num2str(P(seq_P(i,1),1)) ',' num2str(P(seq_P(i,1),2)) ' '
num2str(P(seq_P(i,2),1)) ',' num2str(P(seq_P(i,2),2)) '" fill="none" stroke="black"/>'])
end
seq_Cut1_p1 = [1,2;2,3;3,4;4,1;5,6,;6,7;7,8;8,5;9,10;10,11;11,12;12,9;13,14;14,15;15,16;16,13];
for i = 1:size(seq_Cut1_p1,1)
   disp(['<polyline points="' num2str(Cut1_p1(seq_Cut1_p1(i,1),1)) ','
num2str(Cut1_p1(seq_Cut1_p1(i,1),2)) ' ' num2str(Cut1_p1(seq_Cut1_p1(i,2),1)) ','
num2str(Cut1_p1(seq_Cut1_p1(i,2),2)) '" fill="none" stroke="black"/>'])
end
seq_Cut1_p3 = [1,2;2,3;3,4;4,1;5,6,;6,7;7,8;8,5;9,10;10,11;11,12;12,9;13,14;14,15;15,16;16,13];
for i = 1:size(seq_Cut1_p3,1)
   disp(['<polyline points="' num2str(Cut1_p3(seq_Cut1_p3(i,1),1)) ','
num2str(Cut1_p3(seq_Cut1_p3(i,1),2)) ' ' num2str(Cut1_p3(seq_Cut1_p3(i,2),1)) ','
num2str(Cut1_p3(seq_Cut1_p3(i,2),2)) '" fill="none" stroke="black"/>'])
end
seq_Cut2_p8 =
[1,2;2,3;3,4;4,5;5,6;6,7;7,8;8,9;9,10;10,11;12,13;13,14;14,15;15,16;16,17;17,18;18,19;19,20;20,21
;21,22;23,24;25,26];
for i = 1:size(seq_Cut2_p8,1)
   disp(['<polyline points="' num2str(Cut2_p8(seq_Cut2_p8(i,1),1)) ','
num2str(Cut2_p8(seq_Cut2_p8(i,1),2)) ' ' num2str(Cut2_p8(seq_Cut2_p8(i,2),1)) ','
num2str(Cut2_p8(seq_Cut2_p8(i,2),2)) '" fill="none" stroke="black"/>'])
end
seq_Cut2_p10 =
[1,2;2,3;3,4;4,5;5,6;6,7;7,8;8,9;9,10;10,11;12,13;13,14;14,15;15,16;16,17;17,18;18,19;19,20;20,21
;21,22;23,24;25,26];
for i = 1:size(seq_Cut2_p10,1)
   disp(['<polyline points="' num2str(Cut2_p10(seq_Cut2_p10(i,1),1)) ','
num2str(Cut2_p10(seq_Cut2_p10(i,1),2)) ' ' num2str(Cut2_p10(seq_Cut2_p10(i,2),1)) ','
num2str(Cut2_p10(seq_Cut2_p10(i,2),2)) '" fill="none" stroke="black"/>'])
end
seq_Pattern1_p1 = [1:1:7,8,16:-1:10,17:1:31,32;2:1:8,16,15:-1:9,18:1:32,17]';
for i = 1:size(seq_Pattern1_p1,1)
   disp(['<polyline points="' num2str(Pattern1_p1(seq_Pattern1_p1(i,1),1)) ','
num2str(Pattern1_p1(seq_Pattern1_p1(i,1),2)) ' ' num2str(Pattern1_p1(seq_Pattern1_p1(i,2),1)) ','
num2str(Pattern1_p1(seq_Pattern1_p1(i,2),2)) '" fill="none" stroke="black"/>'])
end
seq_Pattern1_p3 = [1:1:7,8,16:-1:10,17:1:31,32;2:1:8,16,15:-1:9,18:1:32,17]';
for i = 1:size(seq_Pattern1_p3,1)
   disp(['<polyline points="' num2str(Pattern1_p3(seq_Pattern1_p3(i,1),1)) ','
num2str(Pattern1_p3(seq_Pattern1_p3(i,1),2)) ' ' num2str(Pattern1_p3(seq_Pattern1_p3(i,2),1)) ','
num2str(Pattern1_p3(seq_Pattern1_p3(i,2),2)) '" fill="none" stroke="black"/>'])
end
seq_Pattern2_p2 = [1:1:9,11:1:19,21:1:29,31:1:39;2:1:10,12:1:20,22:1:30,32:1:40]';
for i = 1:size(seq_Pattern2_p2,1)
```

```
   disp(['<polyline points="' num2str(Pattern2_p2(seq_Pattern2_p2(i,1),1)) ','
num2str(Pattern2_p2(seq_Pattern2_p2(i,1),2)) ' ' num2str(Pattern2_p2(seq_Pattern2_p2(i,2),1)) ','
num2str(Pattern2_p2(seq_Pattern2_p2(i,2),2)) '" fill="none" stroke="black"/>'])
end
seq_Pattern2_p4 = [1:1:9,11:1:19,21:1:29,31:1:39;2:1:10,12:1:20,22:1:30,32:1:40]';
for i = 1:size(seq_Pattern2_p4,1)
   disp(['<polyline points="' num2str(Pattern2_p4(seq_Pattern2_p4(i,1),1)) ','
num2str(Pattern2_p4(seq_Pattern2_p4(i,1),2)) ' ' num2str(Pattern2_p4(seq_Pattern2_p4(i,2),1)) ','
num2str(Pattern2_p4(seq_Pattern2_p4(i,2),2)) '" fill="none" stroke="black"/>'])
end
for i = 1:4
   disp(['<circle cx="' num2str(Cir_pos_p2(i,1)) '" cy="' num2str(Cir_pos_p2(i,2)) '" r="'
num2str(r) '" fill="none" stroke="black"/>'])
end
for i = 1:4
   disp(['<circle cx="' num2str(Cir_pos_p4(i,1)) '" cy="' num2str(Cir_pos_p4(i,2)) '" r="'
num2str(r) '" fill="none" stroke="black"/>'])
end
seq_Sc = [1:1:11,13:1:23,25:1:35,37:1:47;2:1:12,14:1:24,26:1:36,38:1:48]';
for i = 1:size(seq_Sc,1)
   disp(['<polyline points="' num2str(Sc_p1(seq_Sc(i,1),1)) ',' num2str(Sc_p1(seq_Sc(i,1),2)) ' '
num2str(Sc_p1(seq_Sc(i,2),1)) ',' num2str(Sc_p1(seq_Sc(i,2),2)) '" fill="none"
stroke="black"/>'])
end
for i = 1:size(seq_Sc,1)
   disp(['<polyline points="' num2str(Sc_p3(seq_Sc(i,1),1)) ',' num2str(Sc_p3(seq_Sc(i,1),2)) ' '
num2str(Sc_p3(seq_Sc(i,2),1)) ',' num2str(Sc_p3(seq_Sc(i,2),2)) '" fill="none"
stroke="black"/>'])
end
fprintf('<image href="icon-logo.png" x="%f" y="%f" width="%fpx"
transform="translate(%f,%f)"/>\n',0.5*L,H/3,H/3,-H/3/2,-H/3/2) % icon-logo.png: 234*234 pixels
fprintf('<text x="%f" y="%f" font-size="%fem" transform="translate(%f,%f)" font-
family="monospace">Digital Manufacturing</text>', L/2,H/2,H/14/16,-H/16/2*10,-H/14/16/2)
fprintf('<text x="%f" y="%f" font-size="%fem" transform="translate(%f,%f)" font-
family="monospace">xg2381   tg2776</text>', L/2,2/3*H,H/14/16,-H/16/2*8,-H/14/16/2)
fprintf('<text x="%f" y="%f" font-size="%fem" transform="translate(%f,%f)" font-
family="monospace">Laser Cut Box</text>', L/2,2*H+3/2*W,H/10/16,-H/16/2*9,-H/14/16/2)
disp('</svg>')
diary off
```

```
function
[L,W,H,P,Cut1_p1,Cut1_p3,Cut2_p8,Cut2_p10,Pattern1_p1,Pattern1_p3,Pattern2_p2,Pattern2_p4,Cir_pos
_p2,Cir_pos_p4,Sc_p1,Sc_p3] = box
   T = 3; %thickness: 3 mm / The default unit in svg file is pixel.
   initpos = [0,0];
   while true
       L = input("Input the value of LENGTH (mm):\n");
       W = input("Input the value of WIDTH (mm):\n");
       H = input("Input the value of HEIGHT (mm):\n");
       if (initpos(1) + 2*L + W + 6*T)<= 457 && max(initpos(2) + 2*W + 3*T, initpos(2) + 2*H +
3*T)<= 457
           break;
       else
           disp('Error: Exceed the dismension of the board (457 mm * 304 mm)');
       end
   end
   p1 = initpos;
   p2 = p1 + [0,H];
   p3 = p2 + [0,W];
   p4 = p3 + [0,H];
   p5 = p4 + [0,W];
   p6 = p5 + [L,0];
   p7 = p6 + [0,-W];
```

```
    p8 = p7 + [0,-H];
    p9 = p8 + [0,-W];
    p10 = p9 + [0,-H];
    p11 = p10 + [W,0];
    p12 = p11 + [0,H];
    p13 = p12 + [0,W];
    p14 = p13 + [0,H];
    P = [p1;p2;p3;p4;p5;p6;p7;p8;p9;p10;p11;p12;p13;p14];
    line([p1(1);p5(1)],[p1(2);p5(2)])
    line([p5(1);p6(1)],[p5(2);p6(2)])
    line([p6(1);p10(1)],[p6(2);p10(2)])
    line([p11(1);p12(1)],[p11(2);p12(2)])
    line([p13(1);p14(1)],[p13(2);p14(2)])
    line([p1(1);p11(1)],[p1(2);p11(2)])
    line([p2(1);p12(1)],[p2(2);p12(2)])
    line([p3(1);p13(1)],[p3(2);p13(2)])
    line([p4(1);p14(1)],[p4(2);p14(2)])
    axis equal
    axis([0,L+W,0,2*(W+H)])
    [Cut1_p1] = Cut1(p1,L,W,H,T);
    [Cut1_p3] = Cut1(p3,L,W,H,T);
    [Cut2_p8] = Cut2(p8,L,W,H,T);
    [Cut2_p10] = Cut2(p10,L,W,H,T);
    [Pattern1_p1] = Pattern1(p1,L,W,H,T);
    [Pattern1_p3] = Pattern1(p3,L,W,H,T);
    [Pattern2_p2] = Pattern2(p2,L,W,H,T);
    [Pattern2_p4] = Pattern2(p4,L,W,H,T);
    r = 1.2;
    Cir_pos_p2 = circle(p2,r,L,W,H,T);
    Cir_pos_p4 = circle(p4,r,L,W,H,T);
    Sc_p1 = screw(p1,L,W,H,T);
    Sc_p3 = screw(p3,L,W,H,T);
 end
function [C] = Cut1(po,L,W,H,T)
    c1 = po + [0,3/8*H];
    c2 = c1 + [0,-H/8];
    c3 = c2 + [T,0];
    c4 = c3 + [0,H/8];
    Cpart1 = [c1;c2;c3;c4];
    line(Cpart1(:,1),Cpart1(:,2))
    line([c1(1),c4(1)],[c1(2),c4(2)])
    c5 = po + [0,5/8*H];
    c6 = c5 + [0,-H/8];
    c7 = c6 + [T,0];
    c8 = c7 + [0,H/8];
    Cpart2 = [c5;c6;c7;c8];
    line(Cpart2(:,1),Cpart2(:,2))
    line([c5(1),c8(1)],[c5(2),c8(2)])

    c9 = po + [L,5/8*H];
    c10 = c9 + [0,-H/8];
    c11 = c10 + [-T,0];
    c12 = c11 + [0,H/8];
    Cpart3 = [c9;c10;c11;c12];
    line(Cpart3(:,1),Cpart3(:,2))
    line([c9(1),c12(1)],[c9(2),c12(2)])
    c13 = po + [L,3/8*H];
    c14 = c13 + [0,-H/8];
    c15 = c14 + [-T,0];
    c16 = c15 + [0,H/8];
    Cpart4 = [c13;c14;c15;c16];
    line(Cpart4(:,1),Cpart4(:,2))
    line([c13(1),c16(1)],[c13(2),c16(2)])
```

```
   C = [Cpart1;Cpart2;Cpart3;Cpart4];
end
function [C] = Cut2(po,L,W,H,T)
   C_init = [];
   C_init(1,:) = po + [T,0];
   C_init(2,:) = C_init(1,:) + [0,1/4*H];
   C_init(3,:) = C_init(2,:) + [-T,0];
   C_init(4,:) = C_init(3,:) + [0,H/8];
   C_init(5,:) = C_init(4,:) + [T,0];
   C_init(6,:) = C_init(5,:) + [0,H/8];
   C_init(7,:) = C_init(6,:) + [-T,0];
   C_init(8,:) = C_init(7,:) + [0,H/8];
   C_init(9,:) = C_init(8,:) + [T,0];
   C_init(10,:) = po + [T,H];
   C_init(11,:) = po + [0,H];
   line(C_init(:,1),C_init(:,2))
   %%%% Vmirror %%%%%%
   C_vm = [2*po(1) + W - C_init(:,1),C_init(:,2)];
   line(C_vm(:,1),C_vm(:,2))
   %%%% others %%%%%%%
   c1 = po + [T,T];
   c2 = po + [W - T,T];
   line([c1(1),c2(1)],[c1(2),c2(2)])
   c3 = po + [T,H - T];
   c4 = po + [W - T,H - T];
   line([c3(1),c4(1)],[c3(2),c4(2)])
   C = [C_init;C_vm;c1;c2;c3;c4];
end
function [Pattern_po] = Pattern1(po,L,W,H,T)
   c1 = po + [T/6,0];
   c2 = c1 + [0,T];
   c3 = po + [L/2-2*T/3,T];
   c4 = c3 - [0,T];
   c5 = po + [L/2-T/3,0];
   c6 = c5 + [0,T];
   c7 = po + [L/2-T/6,T];
   c8 = po + [L/2-T/6,0];
   %%% mirroring vertical  %%%%%
   c9 = [L - c1(1),c1(2)];
   c10 = [L - c2(1),c2(2)];
   c11 = [L - c3(1),c3(2)];
   c12 = [L - c4(1),c4(2)];
   c13 = [L - c5(1),c5(2)];
   c14 = [L - c6(1),c6(2)];
   c15 = [L - c7(1),c7(2)];
   c16 = [L - c8(1),c8(2)];
   Cpart1 = [c1;c2;c3;c4;c5;c6;c7;c8;c9;c10;c11;c12;c13;c14;c15;c16];
   line(Cpart1(:,1),Cpart1(:,2))
   line([c1(1),c8(1)],[c1(2),c8(2)])

   %%% mirroring horizonal %%%%%
   c17 = [c1(1),2*po(2) + H - c1(2)];
   c18 = [c2(1),2*po(2) + H - c2(2)];
   c19 = [c3(1),2*po(2) + H - c3(2)];
   c20 = [c4(1),2*po(2) + H - c4(2)];
   c21 = [c5(1),2*po(2) + H - c5(2)];
   c22 = [c6(1),2*po(2) + H - c6(2)];
   c23 = [c7(1),2*po(2) + H - c7(2)];
   c24 = [c8(1),2*po(2) + H - c8(2)];
   c25 = [c9(1),2*po(2) + H - c9(2)];
   c26 = [c10(1),2*po(2) + H - c10(2)];
   c27 = [c11(1),2*po(2) + H - c11(2)];
   c28 = [c12(1),2*po(2) + H - c12(2)];
```

```
    c29 = [c13(1),2*po(2) + H - c13(2)];
    c30 = [c14(1),2*po(2) + H - c14(2)];
    c31 = [c15(1),2*po(2) + H - c15(2)];
    c32 = [c16(1),2*po(2) + H - c16(2)];
    Cpart2 = [c17;c18;c19;c20;c21;c22;c23;c24;c25;c26;c27;c28;c29;c30;c31;c32];
    line(Cpart2(:,1),Cpart2(:,2))
    line([c17(1),c32(1)],[c17(2),c32(2)])
    Pattern_po = [Cpart1;Cpart2];
end
function [Pattern_po] = Pattern2(po,L,W,H,T)
    c1 = po + [0,T];
    c2 = c1 + [T/6,0];
    c3 = c2 - [0,T];
    c4 = po + [L/2-2/3*T,0];
    c5 = c4 + [0,T];
    c6 = po + [L/2-T/3,T];
    c7 = po + [L/2-T/3,0];
    c8 = po + [L/2-T/6,0];
    c9 = c8 + [0,T];
    c10 = c9 + [T/6,0];
    Cpart1 = [c1;c2;c3;c4;c5;c6;c7;c8;c9;c10];
    line(Cpart1(:,1),Cpart1(:,2))
    %%% mirroring vertical  %%%%%
    c11 = [2*po(1) + L - c1(1),c1(2)];
    c12 = [2*po(1) + L - c2(1),c2(2)];
    c13 = [2*po(1) + L - c3(1),c3(2)];
    c14 = [2*po(1) + L - c4(1),c4(2)];
    c15 = [2*po(1) + L - c5(1),c5(2)];
    c16 = [2*po(1) + L - c6(1),c6(2)];
    c17 = [2*po(1) + L - c7(1),c7(2)];
    c18 = [2*po(1) + L - c8(1),c8(2)];
    c19 = [2*po(1) + L - c9(1),c9(2)];
    c20 = [2*po(1) + L - c10(1),c10(2)];
    Cpart2 = [c11;c12;c13;c14;c15;c16;c17;c18;c19;c20];
    line(Cpart2(:,1),Cpart2(:,2))
    %%% mirroring horizonal %%%%%
    c21 = [c1(1),2*po(2) + W - c1(2)];
    c22 = [c2(1),2*po(2) + W - c2(2)];
    c23 = [c3(1),2*po(2) + W - c3(2)];
    c24 = [c4(1),2*po(2) + W - c4(2)];
    c25 = [c5(1),2*po(2) + W - c5(2)];
    c26 = [c6(1),2*po(2) + W - c6(2)];
    c27 = [c7(1),2*po(2) + W - c7(2)];
    c28 = [c8(1),2*po(2) + W - c8(2)];
    c29 = [c9(1),2*po(2) + W - c9(2)];
    c30 = [c10(1),2*po(2) + W - c10(2)];
    Cpart3 = [c21;c22;c23;c24;c25;c26;c27;c28;c29;c30];
    line(Cpart3(:,1),Cpart3(:,2))
    c31 = [c11(1),2*po(2) + W - c11(2)];
    c32 = [c12(1),2*po(2) + W - c12(2)];
    c33 = [c13(1),2*po(2) + W - c13(2)];
    c34 = [c14(1),2*po(2) + W - c14(2)];
    c35 = [c15(1),2*po(2) + W - c15(2)];
    c36 = [c16(1),2*po(2) + W - c16(2)];
    c37 = [c17(1),2*po(2) + W - c17(2)];
    c38 = [c18(1),2*po(2) + W - c18(2)];
    c39 = [c19(1),2*po(2) + W - c19(2)];
    c40 = [c20(1),2*po(2) + W - c20(2)];
    Cpart4 = [c31;c32;c33;c34;c35;c36;c37;c38;c39;c40];
    line(Cpart4(:,1),Cpart4(:,2))
    Pattern_po = [Cpart1;Cpart2;Cpart3;Cpart4];
end
function pos = circle(po,r,L,W,H,T)
```

```
   pos = zeros(4,2);
   pos(1,:) = po + [L/4-T/4,T/2];
   pos(2,:) = po + [L/4-T/4,W - T/2];
   pos(3,:) = po + [3*L/4+T/4,T/2];
   pos(4,:) = po + [3*L/4+T/4,W - T/2];
end
function [Sc] = screw(po,L,W,H,T)
   Ls = 7; % original nail 10mm then minus thickness 3mm
   Rs = 1.2;
   Wh = 2;
   Lh = 5;
   b = Lh/2 - Rs;
   c1 = po + [L/4-T/4-Rs,T];
   c2 = c1 + [0,Ls/4];
   c3 = c2 + [-b,0];
   c4 = c3 + [0,Wh];
   c5 = c4 + [b,0];
   c6 = c1 + [0,Ls];
   c12 = c1 + [2*Rs,0];
   c11 = c12 + [0,Ls/4];
   c10 = c11 + [b,0];
   c9 = c10 + [0,Wh];
   c8 = c9 + [-b,0];
   c7 = c6 + [2*Rs,0];
   C_init = [c1;c2;c3;c4;c5;c6;c7;c8;c9;c10;c11;c12];
   line(C_init(:,1),C_init(:,2))
   %%% mirroring vertical  %%%%%
   C_vm = [2*po(1) + L - C_init(:,1),C_init(:,2)];
   line(C_vm(:,1),C_vm(:,2))
   %%% mirroring horizonal %%%%%
   Sc= [C_init;C_vm];
   C_hm = [Sc(:,1),2*po(2) + H - Sc(:,2)];
   line(C_hm(1:size(C_hm)/2,1),C_hm(1:size(C_hm)/2,2))
   line(C_hm(size(C_hm)/2+1:end,1),C_hm(size(C_hm)/2+1:end,2))
   Sc= [Sc;C_hm];
end
```

Grading Rubrics

Maximum grade is 100 points. You can accumulate points as follows:

1. **10pt Cover page correct and complete (Page 1)**
2. **10pt small rectangle cut correctly (SVG+Photo) submitted a week before deadline (Page 2)**
3. **10pt Software inputs length, width and height of box and detects errors (show input dialog) (Page 3)**
4. **10pt A description of the software you wrote – calculation steps, formulas, conditions. (Page 3)**
5. **20pt Four examples showing input parameters and output flat pattern. (Page 4-7)**
6. **20pt Printout of the two SVG files and photo of boxes they produced (Page 8-9)**
7. **10pt User-specified text engraved on the top of the box (Page 8-9)**
8. **10pt User-specified text engraved on the front of the box (Page 8-9)**
9. **10pt Columbia logo and "Digital Manufacturing" engraved on the front of the box (Page 8-9)**
10. **10pt A fractal engraving generated by the program embedded on one side of the box**
11. **10pt a lid to cover the box**
12. **10pt a divider that inserts into the box at a user-specified position (e.g. center)**
13. **10pt Any additional parameter accommodated by software, e.g. slanted angle**
14. **10pt box image posted on Ed at least 24h day before the deadline (show screenshot) (Page 10)**
15. **10pt Edited video of the entire process: Keying in parameters, laser cutting, folding, final box in use. (Pending)**