Ingenieur voor een dag


DND
LAB


Speedy100™
trotec®

elektronica-ict | bouwkunde | chemie | elektromechanica

elektronica-ict | **bouwkunde** | chemie | elektromechanica

elektronica-ict | bouwkunde | **chemie** | elektromechanica

elektronica-ict | bouwkunde | chemie | **elektromechanica**

GEAR GENERATOR

Animation: Start/Stop | Freeze | Reset

Speed (RPM)*: 6

* Shift + Enter: Set RPM of the selected gear

Gears: Add New | Remove | Clear

#0 - ratio: 1:1 - RPM: 6

#1 - ratio: 2:1 - RPM: 3

#2 - ratio: 2:1 - RPM: 3

#3 - ratio: 10:1 - RPM: 0.6

Connection properties

Parent gear #: -1 Select

Axle connection: ☑

Connection angle: -90 – +

Gear properties

Number of teeth* (N): 8 – +

Pitch diameter* (D): 2

Diametral pitch (P): 4

Pressure Angle (PA): 27 – +

* Shift + Enter: modifies the Diametral pitch

Free download

Gear CAD file (Beta*): Download DXF

Gear vector image: Download SVG

Gearset vector image: Download Gearset SVG

* DXF download is in BETA status, this means all feedbacks are welcome, to make it a better feature, thanks!

Display

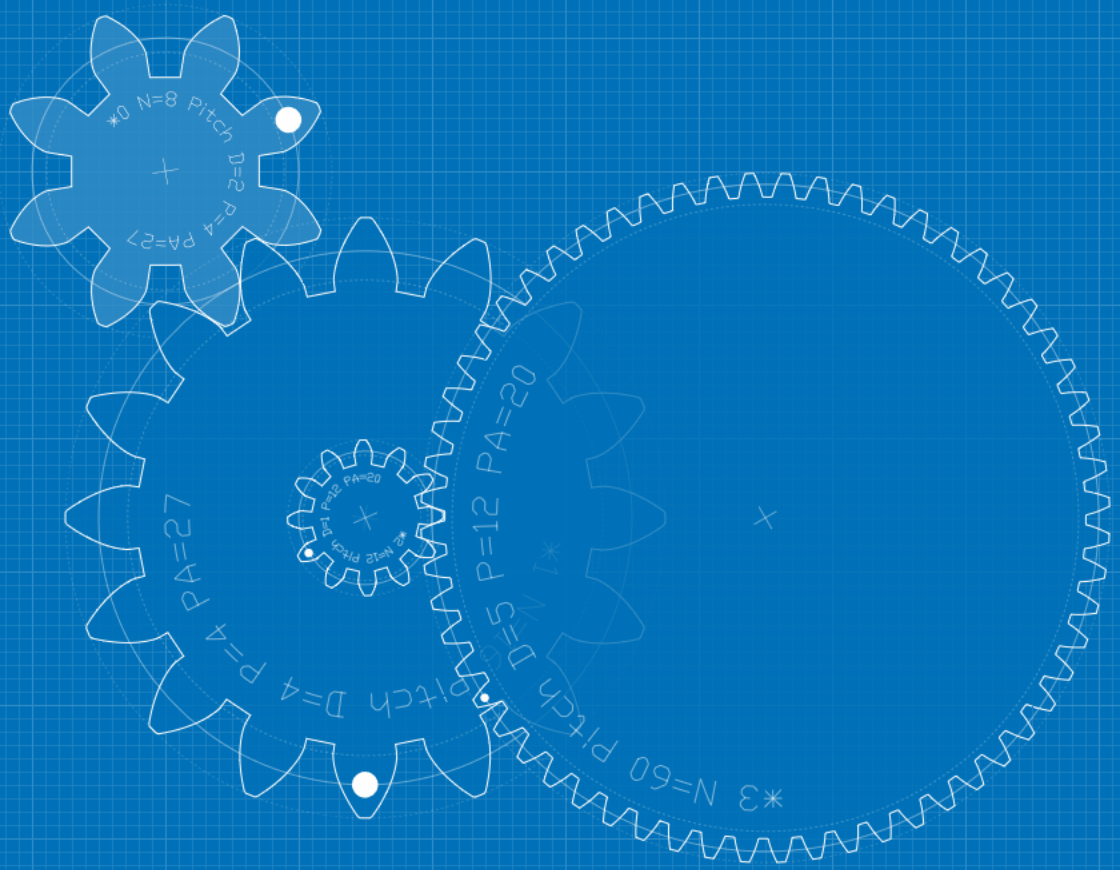
Scale (Pixel per Unit): 100 – +

Grid: ☑

Gear guides: ☑

Gear label: ☑

Color theme: Light | Blueprint | Dark



SVGnest

Open Source nesting

Demo | Upload SVG | Github | FAQ

Start Nest

Download SVG

Placement progress

4%

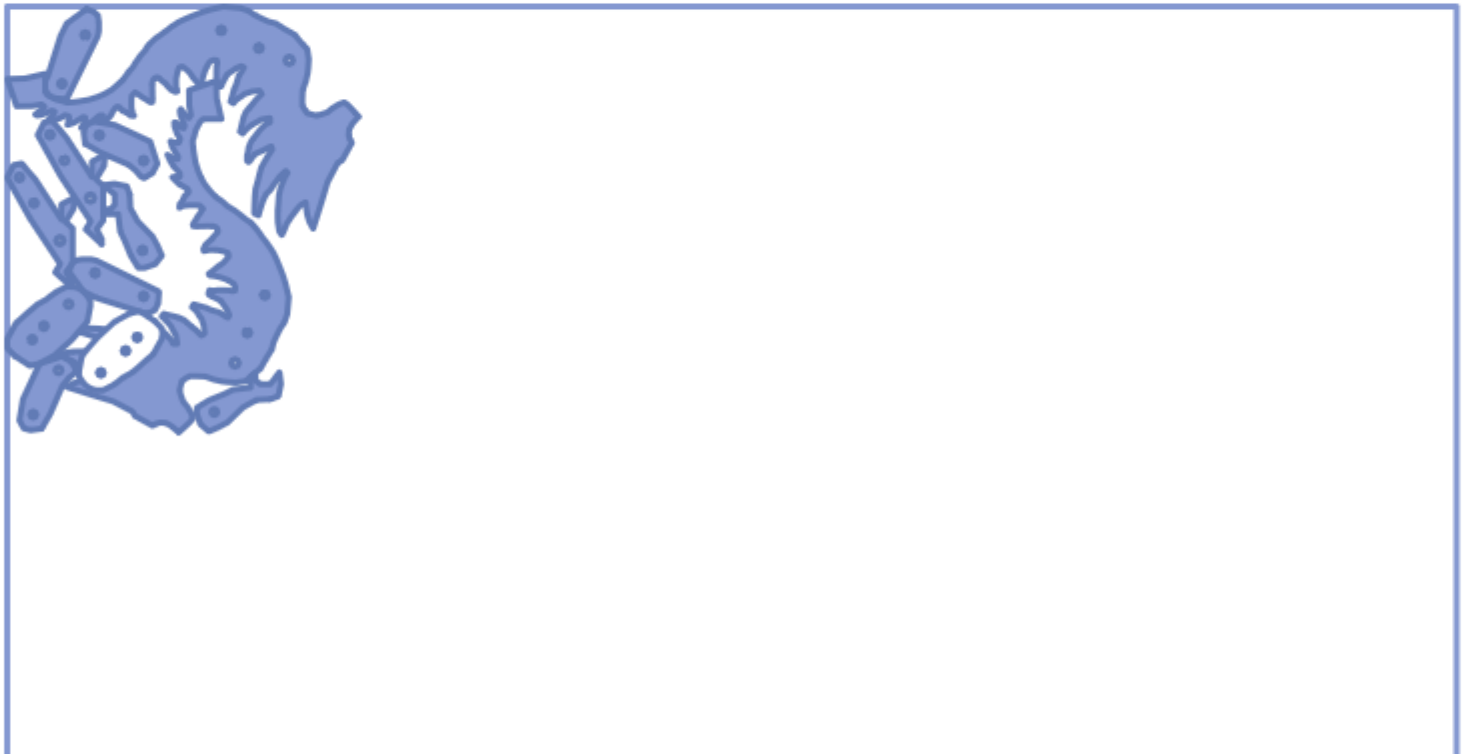
Material Utilization

6

Iterations

11/11

Parts placed

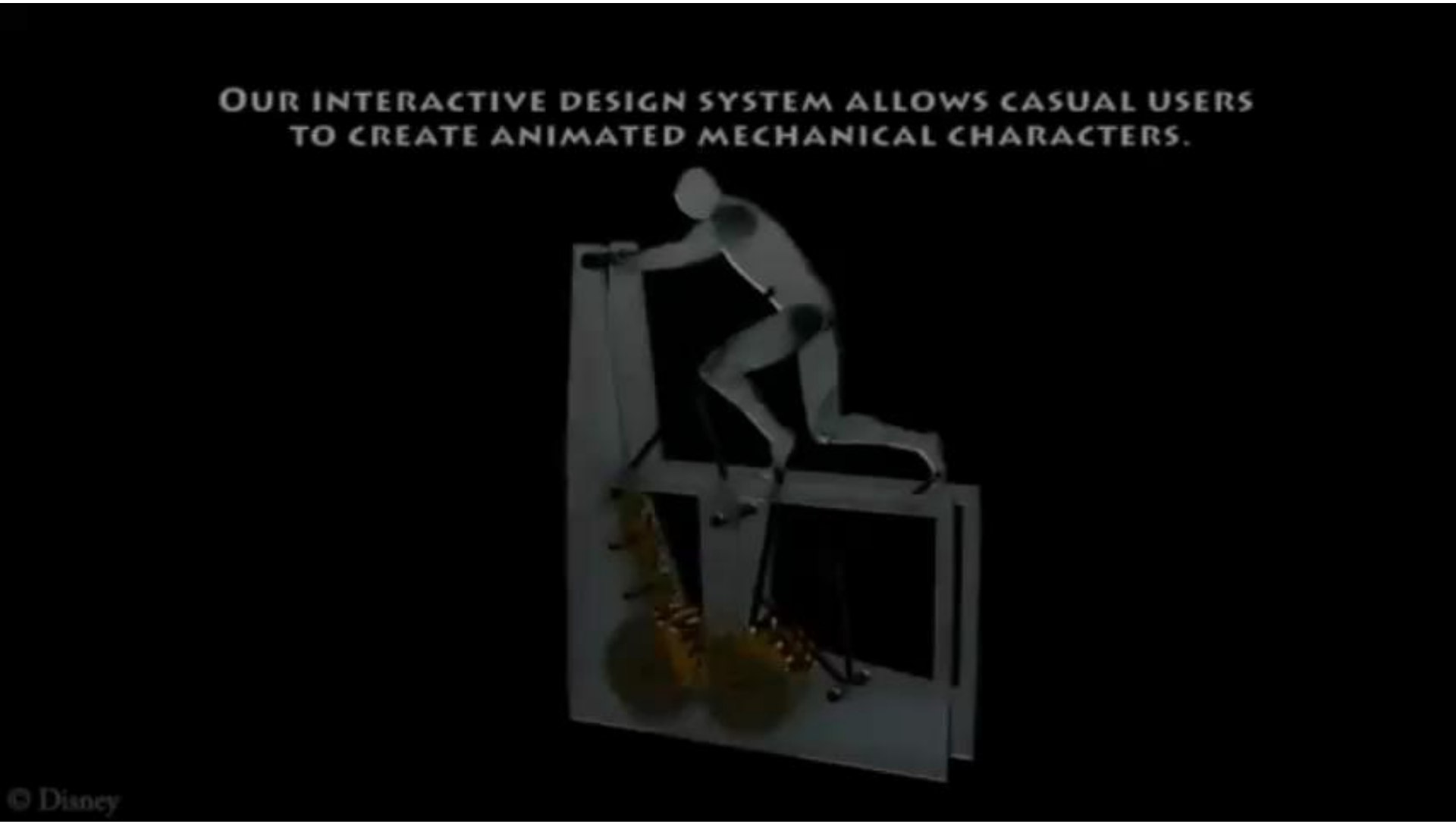


Reverse engineering

Het zijn boeiende tijden

Niet op z’n minst met engineering skills

elektronica-ict | bouwkunde | chemie | elektromechanica


OUR INTERACTIVE DESIGN SYSTEM ALLOWS CASUAL USERS
TO CREATE ANIMATED MECHANICAL CHARACTERS.
© Disney

Computational Design of Mechanical Characters

Stelian Coros*[1] Bernhard Thomaszewski*[1] Gioacchino Noris[1] Shinjiro Sueda[2] Moira Forberg[2]
Robert W. Sumner[1] Wojciech Matusik[3] Bernd Bickel[1]

[1]Disney Research Zurich [2]Disney Research Boston [3]MIT CSAIL

Figure 1: *The interactive design system we introduce allows non-expert users to create complex, animated mechanical characters.*

Abstract

We present an interactive design system that allows non-expert users to create animated mechanical characters. Given an articulated character as input, the user iteratively creates an animation by sketching motion curves indicating how different parts of the character should move. For each motion curve, our framework creates an optimized mechanism that reproduces it as closely as possible. The resulting mechanisms are attached to the character and then connected to each other using gear trains, which are created in a semi-automated fashion. The mechanical assemblies generated with our system can be driven with a single input driver, such as a hand-operated crank or an electric motor, and they can be fabricated using rapid prototyping devices. We demonstrate the versatility of our approach by designing a wide range of mechanical characters, several of which we manufactured using 3D printing. While our pipeline is designed for characters driven by planar mechanisms, significant parts of it extend directly to non-planar mechanisms, allowing us to create characters with compelling 3D motions.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

Keywords: mechanical characters, animation, fabrication, interactive design

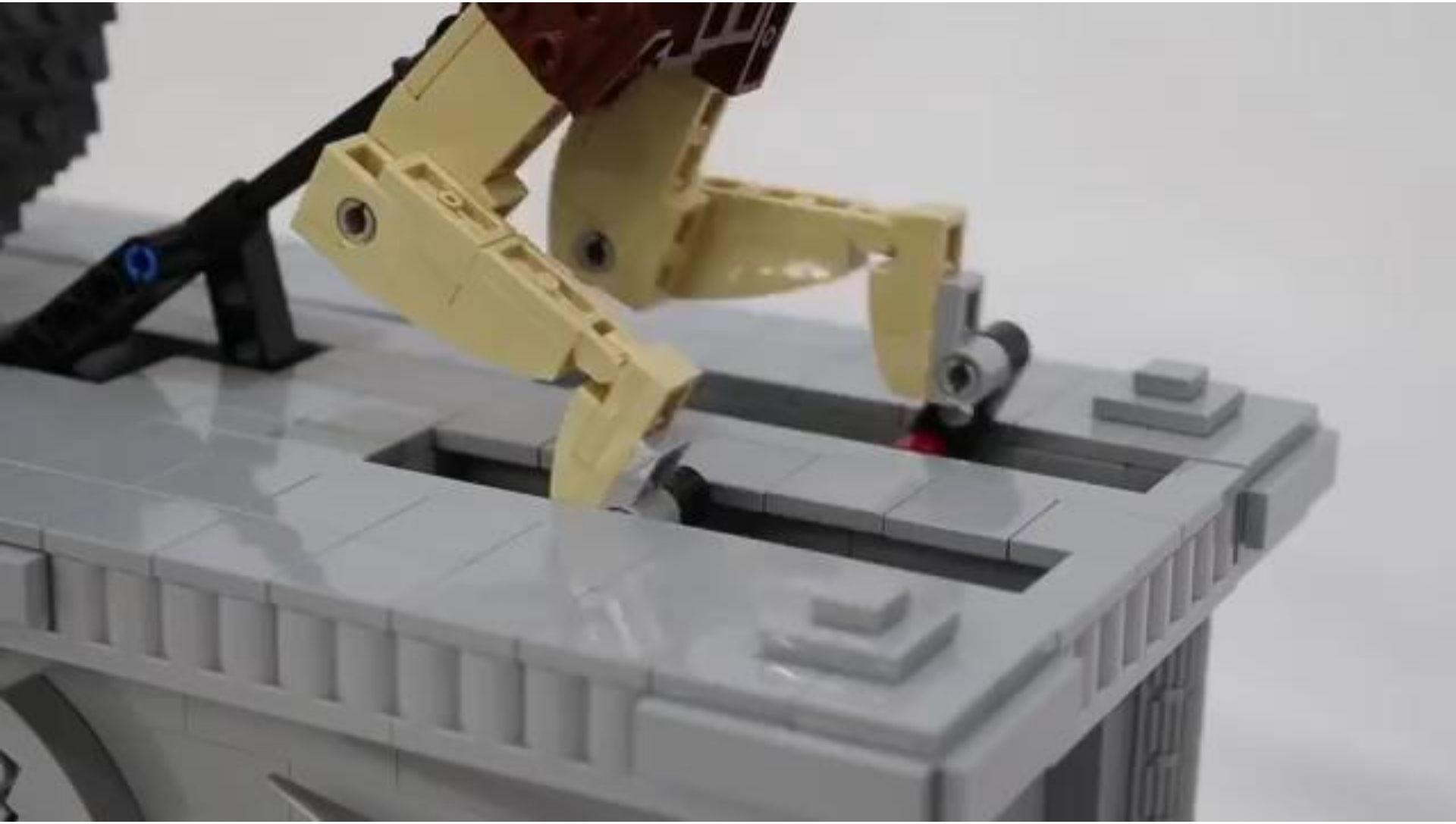
Links: DL PDF

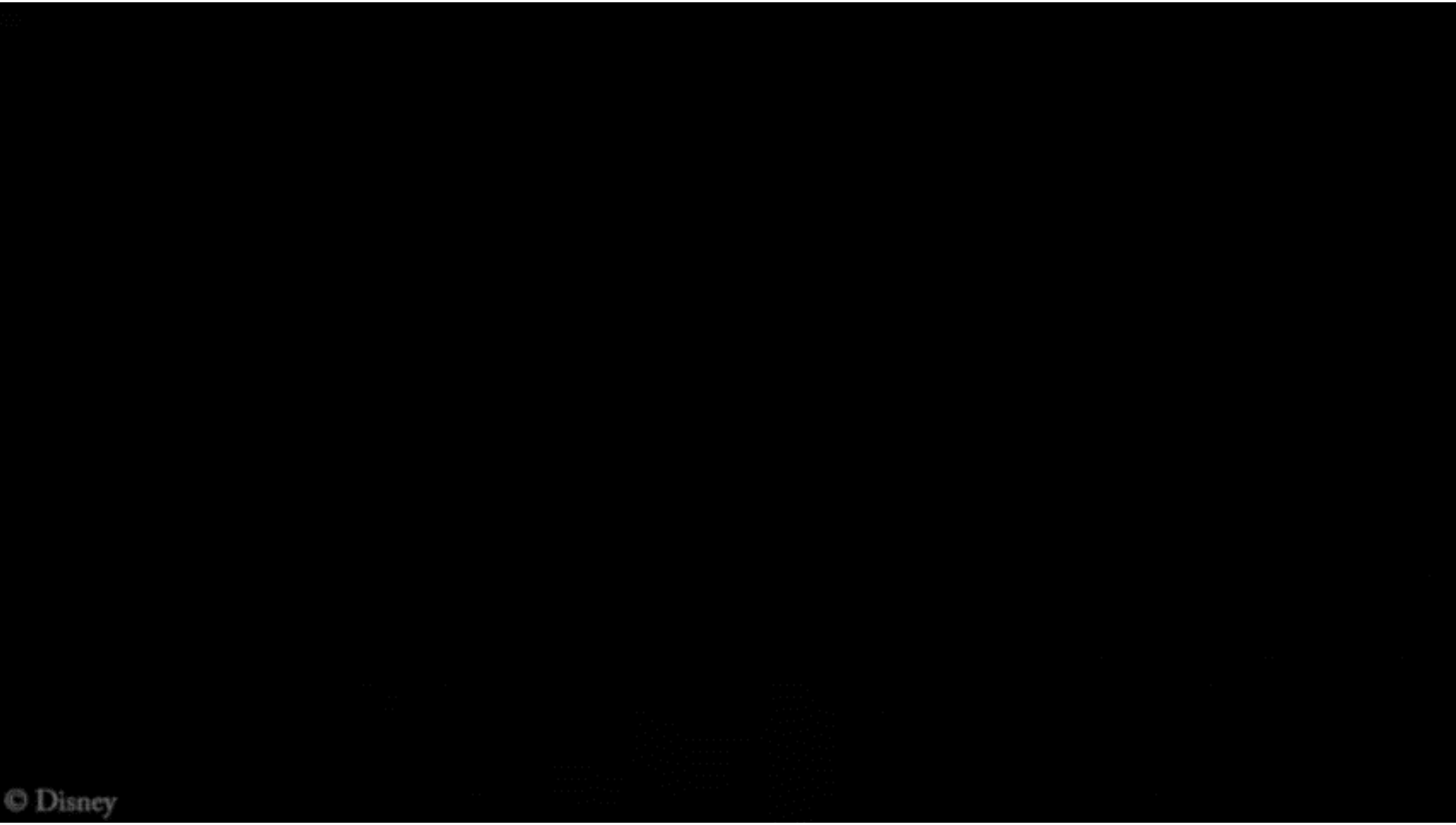
1 Introduction

Character animation allows artists to bring fictional characters to life as virtual actors in animated movies, video games, and live-action films. Well-established software packages assist artists in realizing their creative vision, making almost any digital character and movement possible. In the physical world, animatronic figures play an equivalent role in theme parks and as special effects in movies and television. While these sophisticated robots are far from becoming household items, toys that exhibit mechanical movement are extremely popular as consumer products. However, unlike virtual characters, creating complex and detailed movement for *mechanical characters*, whose motion is determined by physical assemblies of gears and linkages, remains an elusive and challenging task. Although mechanical characters have been part of the toy industry since the nineteenth century [Peppe 2002], design technology for these characters has changed little and is limited to expert designers and engineers. Even for them, the design process is largely trial and error, with many iterations needed to produce an acceptable result. Since iteration times increase greatly as the complexity of the design space increases, mechanical characters are limited in scope and complexity, which in turn limits the range of possible movement and the creative freedom of the designers.

We present a computational design system that allows non-expert users to design and fabricate complex animated mechanical characters (Fig. 1). Our system automates tedious and difficult steps in the design process, and the resulting mechanical characters can be fabricated using rapid manufacturing methods such as 3D printing. Interactivity is a core design principle of our system, allowing users to quickly explore many different mechanical design options, as the motion of the characters is iteratively created.

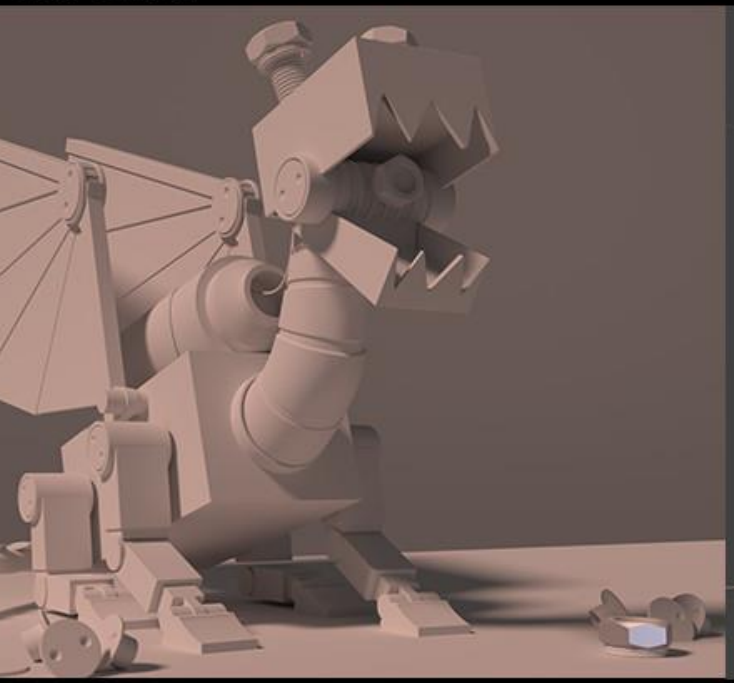
In order to make the computational design problem tractable, we limit the scope of this work to characters that perform cyclic mo-



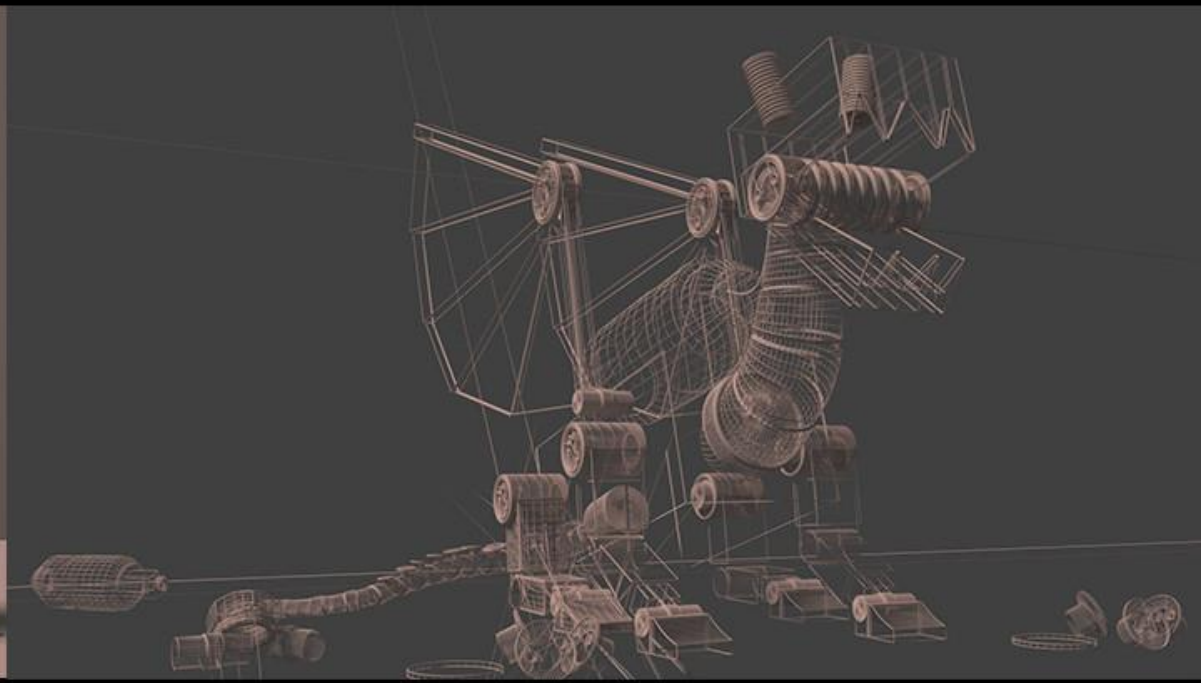

© Disney

COMPOSITION

SHADING

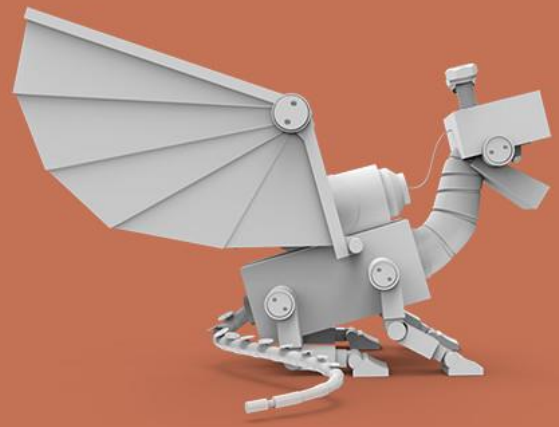


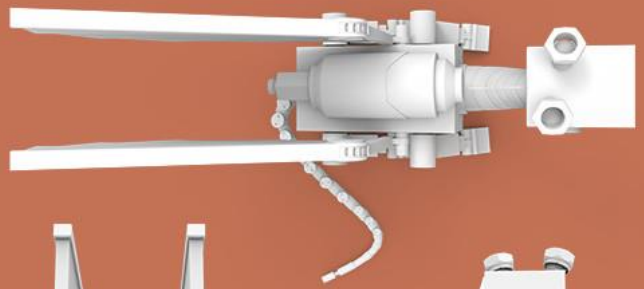
DIFFUSE

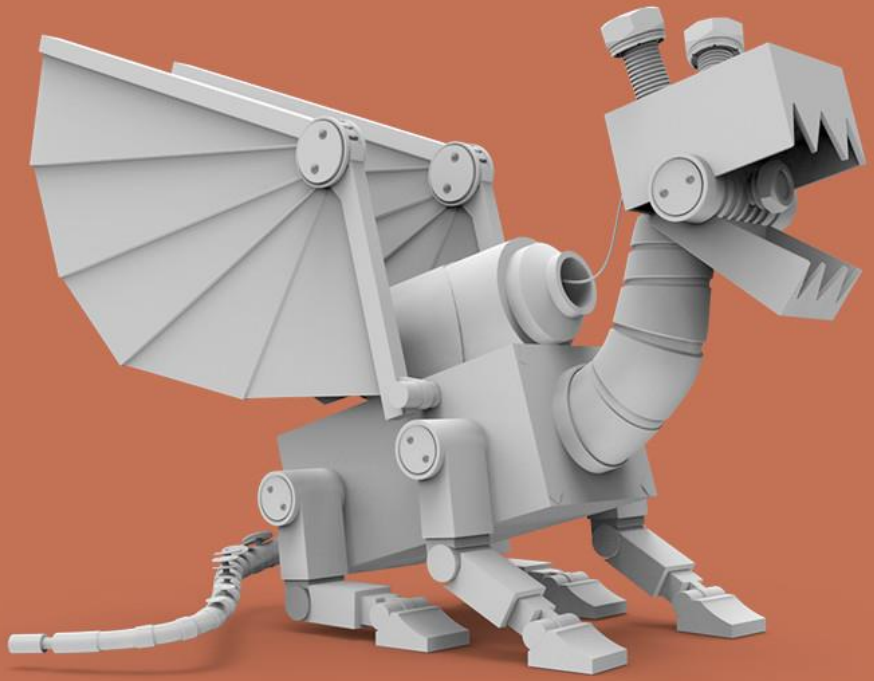


WIREFRAME


TINY DRAGON
GERARD PASQUAL





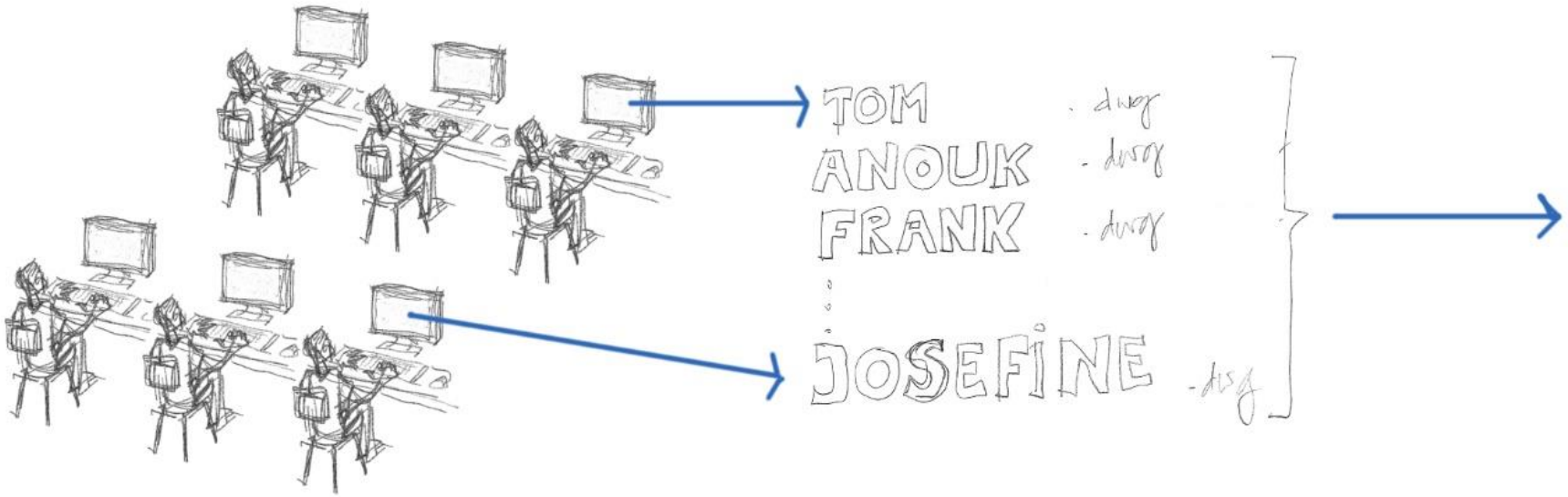


TINY DRAGON DIFUSSE

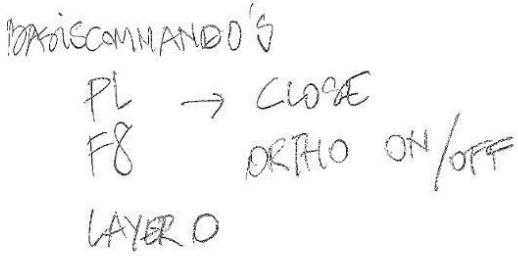
GERARD PASQUAL


GODZILLA
怪獣惑星

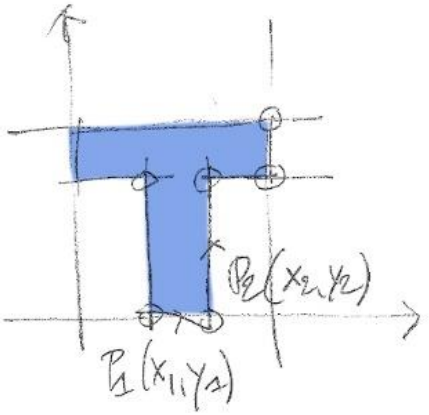
cad



cad



ecologie!



→ alle kamers (dwg) → EXTRAHEER COÖRDINATEN

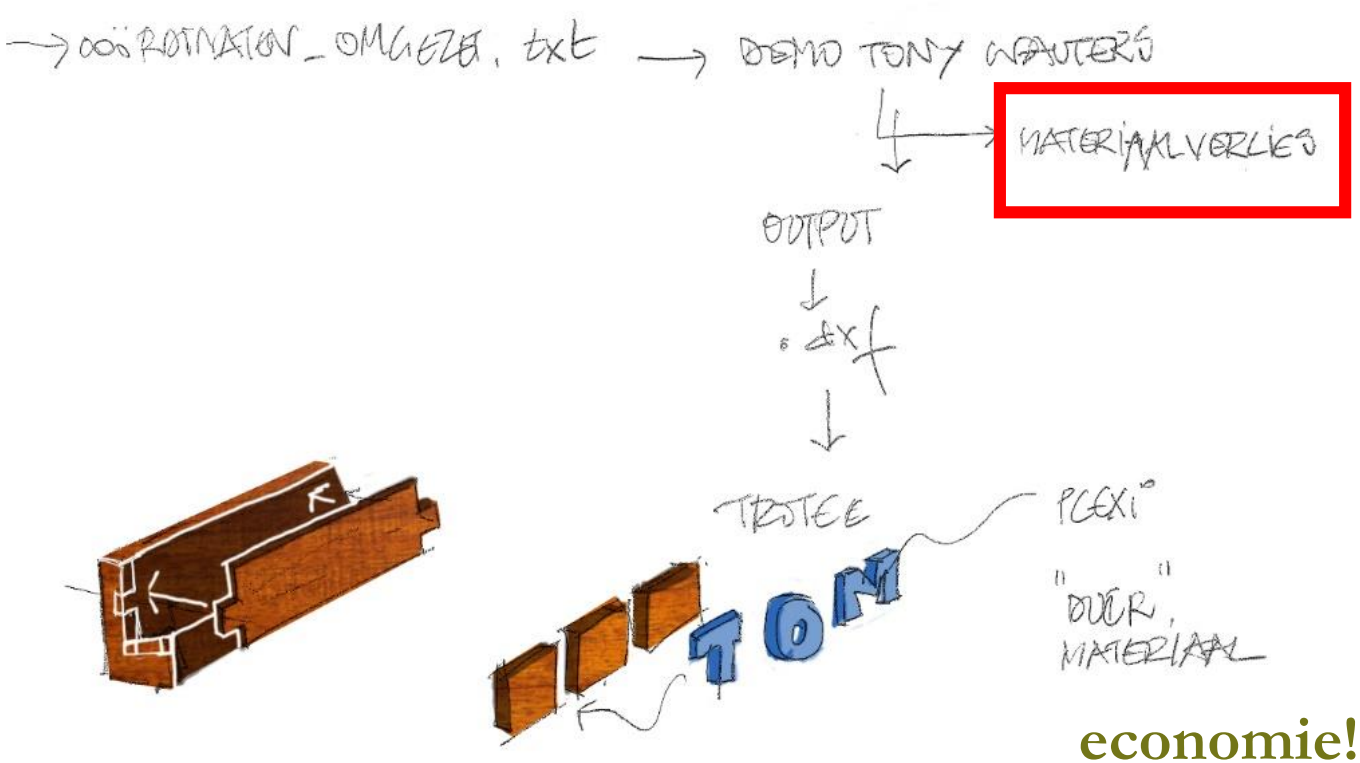
LIST

$P_1(x_1, y_1)$

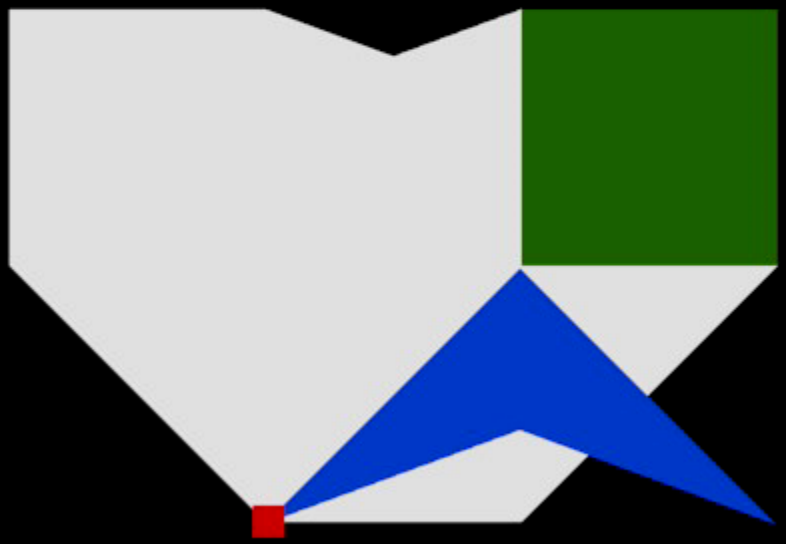
⋮

$P_n(x_n, y_n)$

COORDINATEN.EXE → ECLIPSE


ROTATION_OMGEZET.txt
DEMO TONY WAUTERS
MATERIAALVERLIES
OUTPUT
.dxf
TROTEC
PLEXI
DUUR MATERIAAL
TOM

economie!

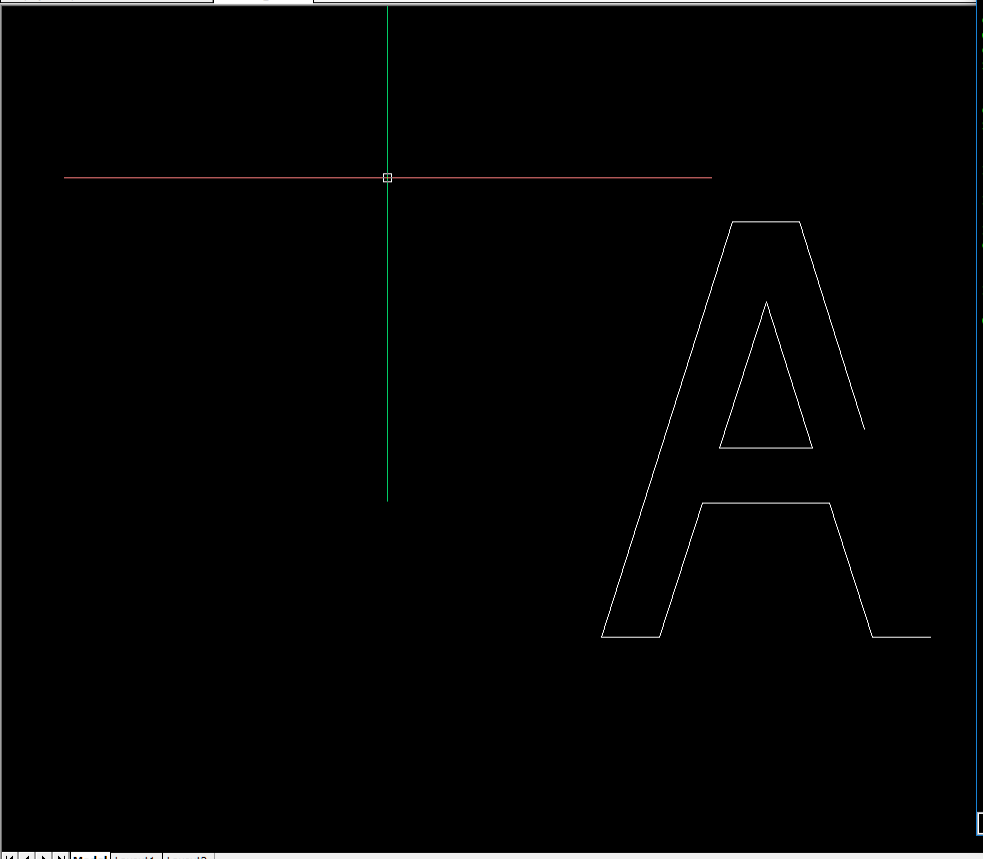


BricsCAD Platinum (ACADEMIC License) - [Drawing1]

File Edit View Insert Settings Tools Draw Dimension Modify Parametric Window Help

```
Location:  X=   296.03  Y=   236.95  Z=   0
Location:  X=   319.63  Y=   162.51  Z=   0
:
```

polylineExporterenNaarPunten | Drawing1*



Model | Layout1 | Layout2

Ready

Standard | ISO-25 | 2D Drafting | SNAP | GRID | ORTHO | POLAR | ESNAP | STRACK | LWT | TILE | DUCS | DYN | QUAD | RT | TIPS | None

BricsCAD Prompt History

```
:
: CHAMFER
Chamfer (dist1=0, dist2=0) Select first entity [chamfer Settings.../Polyline/Angle/Distance/mEthod/Trim/Undo/Multiple]:
Select second entity (select with pressed SHIFT to make corner):
:
: CHAMFER
Chamfer (dist1=0, dist2=0) Select first entity [chamfer Settings.../Polyline/Angle/Distance/mEthod/Trim/Undo/Multiple]:
Opposite Corner:
Chamfer (dist1=0, dist2=0) Select first entity [chamfer Settings.../Polyline/Angle/Distance/mEthod/Trim/Undo/Multiple]:
Select second entity (select with pressed SHIFT to make corner):
:
: CHAMFER
Chamfer (dist1=0, dist2=0) Select first entity [chamfer Settings.../Polyline/Angle/Distance/mEthod/Trim/Undo/Multiple]:
Select second entity (select with pressed SHIFT to make corner):
: ZE
: z
Zoom [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale (nx/nxp)/Window/OBject] <Scale (nX/nXP)>:enil
: Z
Zoom [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale (nx/nxp)/Window/OBject] <Scale (nX/nXP)>:OP
Please try again.
Zoom [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale (nx/nxp)/Window/OBject] <Scale (nX/nXP)>:
Cancel
: ZO
: z
Zoom [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale (nx/nxp)/Window/OBject] <Scale (nX/nXP)>:onil
:
Opposite corner:
: LI
Entities in set: 2
---------- Lwpolyline ------------------------------------------------
                  Handle:  88
           Current space:  Model
                   Layer:  0
                   Color:  256 (BYLAYER)
                Linetype:  ByLayer
         Polyline Flags:  Open
                    Area:  2925.69
               Perimeter:  188.37
                Location:  X=   225.04  Y=   88.06  Z=   0
                Location:  X=   246.02  Y=   88.06  Z=   0
                Location:  X=   261.21  Y=   136  Z=   0
                Location:  X=   307.05  Y=   136  Z=   0
                Location:  X=   322.25  Y=   88.06  Z=   0
                Location:  X=   343.23  Y=   88.06  Z=   0

---------- Lwpolyline ------------------------------------------------
                  Handle:  82
           Current space:  Model
                   Layer:  0
                   Color:  256 (BYLAYER)
                Linetype:  ByLayer
         Polyline Flags:  Open
                    Area:  6170.67
               Perimeter:  258.08
                Location:  X=   225.04  Y=   88.06  Z=   0
                Location:  X=   272.24  Y=   236.95  Z=   0
                Location:  X=   284.13  Y=   236.95  Z=   0
                Location:  X=   296.03  Y=   236.95  Z=   0
                Location:  X=   319.63  Y=   162.51  Z=   0
:
```

9:43
8/02/2018

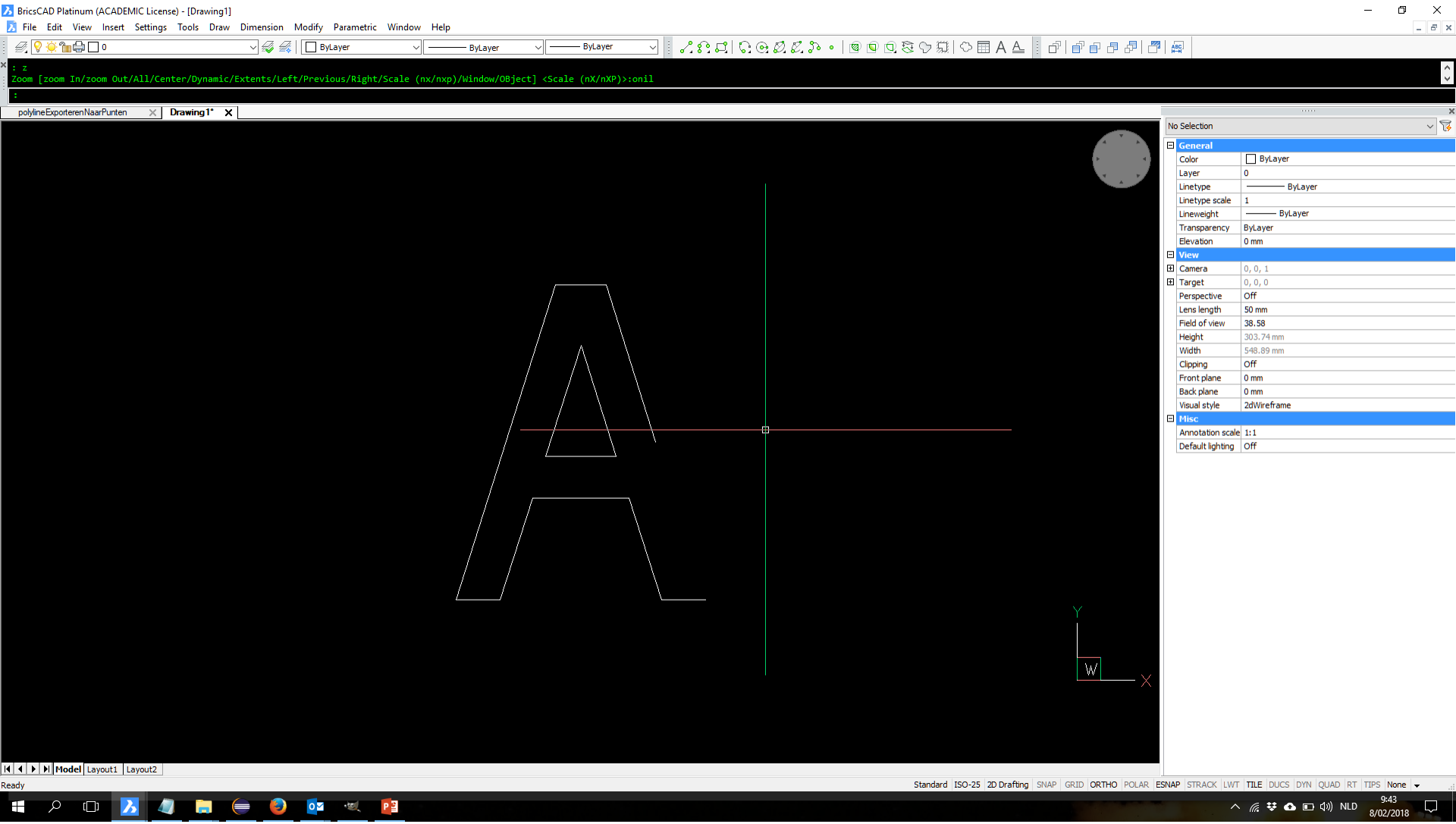
KU LEUVEN

JNFP: a robust and open-source Java based nofit polygon generator library

Tony Wauters, Stiaan Uyttersprot, Eline Esprit

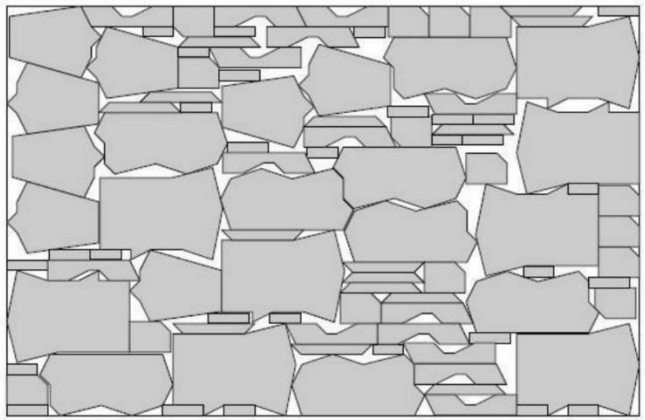
CODeS research group
KU Leuven, Belgium

KU LEUVEN


BricsCAD Platinum (ACADEMIC License) - [Drawing1]
File Edit View Insert Settings Tools Draw Dimension Modify Parametric Window Help
: z
Zoom [zoom In/zoom Out/All/Center/Dynamic/Extents/Left/Previous/Right/Scale (nx/nxp)/Window/OBject] <Scale (nX/nXP)>:onil
polylineExporterenNaarPunten
Drawing1*
No Selection
General
Color ByLayer
Layer 0
Linetype ByLayer
Linetype scale 1
Lineweight ByLayer
Transparency ByLayer
Elevation 0 mm
View
Camera 0, 0, 1
Target 0, 0, 0
Perspective Off
Lens length 50 mm
Field of view 38.58
Height 303.74 mm
Width 548.89 mm
Clipping Off
Front plane 0 mm
Back plane 0 mm
Visual style 2dWireframe
Misc
Annotation scale 1:1
Default lighting Off
Y
W
X
Model Layout1 Layout2
Ready
Standard ISO-25 2D Drafting SNAP GRID ORTHO POLAR ESNAP STRACK LWT TILE DUCS DYN QUAD RT TIPS None
NLD
9:43
8/02/2018

Nesting problems

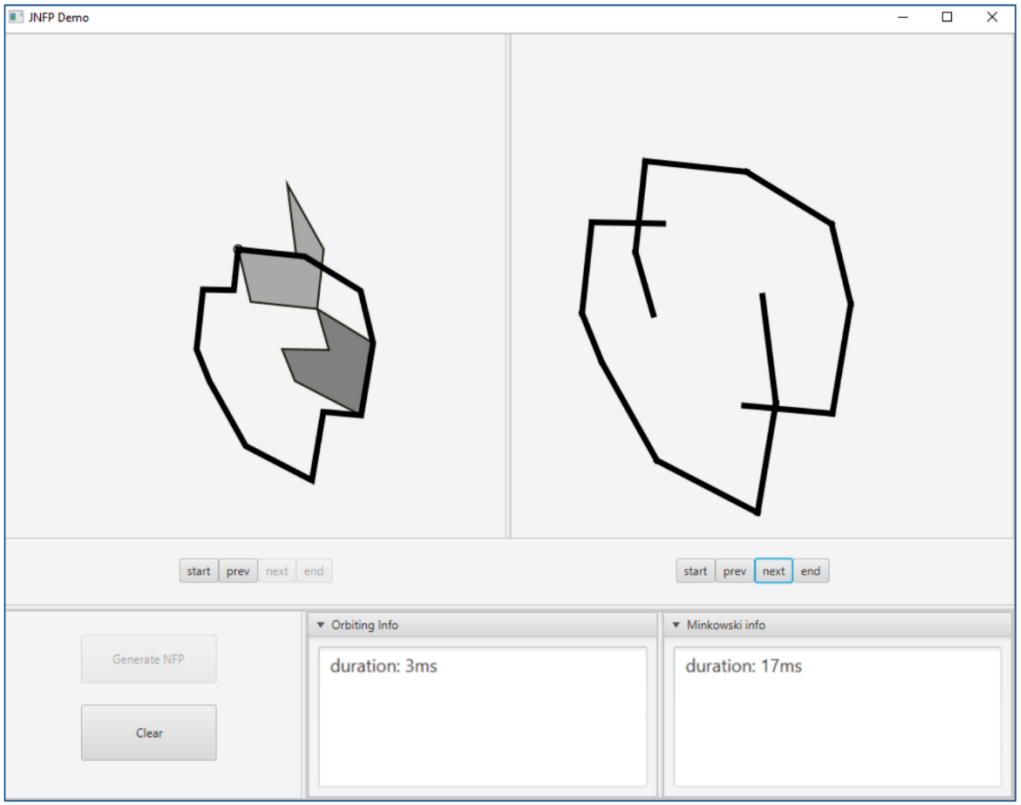
- Cutting and packing problems involving irregular shapes



Shirts*

KU LEUVEN

Demo



KU LEUVEN

Persoon.java | Punt.java | *BestandInlezeraar.java | FileFormatPolygons.txt | gegevensVanuitBricscad.txt

```
:: Count.lsp | Version 1.5 | © Lee Mac 2018 www.lee-mac.com ::
:: "count" - Main Program | "countsettings" - Settings ::
:
Opposite corner:
: LI
Entities in set: 1
---------- Lwpolyline ------------------------------------------
                Handle:  C1
         Current space:  Model
                 Layer:  0
                 Color:  256 (BYLAYER)
             Linetype:  ByLayer
        Polyline Flags:  Open
                  Area:  350000
             Perimeter:  2777.27
              Location:  X=   -11.74  Y=   100  Z=   0
              Location:  X=   188.26  Y=   0  Z=   0
              Location:  X=   388.26  Y=   0  Z=   0
              Location:  X=   488.26  Y=   200  Z=   0
              Location:  X=   588.26  Y=   0  Z=   0
              Location:  X=   988.26  Y=   0  Z=   0
              Location:  X=   988.26  Y=   400  Z=   0
              Location:  X=   588.26  Y=   400  Z=   0
              Location:  X=   488.26  Y=   300  Z=   0
              Location:  X=   388.26  Y=   400  Z=   0
              Location:  X=   188.26  Y=   400  Z=   0
              Location:  X=   -11.74  Y=   300  Z=   0
```

Console | Terminal

<terminated> BestandInlezeraar (Java Ap

```
an example of a polygon contai
2
12
0 100
200 0
400 0
500 200
600 0
1000 0
1000 400
600 400
500 300
400 400
200 400
0 300

4
150 100
150 300
350 300
350 100

4
600 100
600 300
800 300
800 100
```

Outline

An outline is not available.

imiGoesGraphics180207/FileFormatPolygons.txt - Eclipse

Search Project Run Window Help

Type Hierarchy

t package)
tandInlezeraar.java
niGoesGraphics.java
in.java
eel.java
t.java
Polygons.txt
anuitBricscad.txt
ector
ng160426
ng170423
503
IngrediëntenEnIO
rden
tStromen
neren
70323
pp_TEST
etPlaylist
etSongListMargoDhaese
est170418
onderFileChooser
e
e_jokeTorfs
MediaPlayerMetPlaylist_OPGAVE
vens160806
t package)

Persoon.java | Punt.java | *BestandInlezeraar.java | FileFormatPolygons.txt | gegevensVanuitBricscad.txt

```
1  The inputfile for describing polygons has to be formed in the following structure;
2
3  number of holes
4  number of coordinates outer polygon
5  x-coord1 y-coord1
6  x-coord2 y-coord2
7  ....
8  ....
9
10 number of coordinates hole1
11 x-coord1 y-coord1
12 x-coord2 y-coord2
13 ....
14 ....
15
16 number of coordinates hole2
17 ....
18 ....
19
20 an example of a polygon containing two holes:
21 2
22 12
23 0 100
24 200 0
25 400 0
26 500 200
27 600 0
28 1000 0
29 1000 400
30 600 400
31 500 300
32 400 400
33 200 400
34 0 300
35
36 4
37 150 100
38 150 300
39 350 300
40 350 100
41
42 4
43 600 100
44 600 300
45 800 300
46 800 100
```