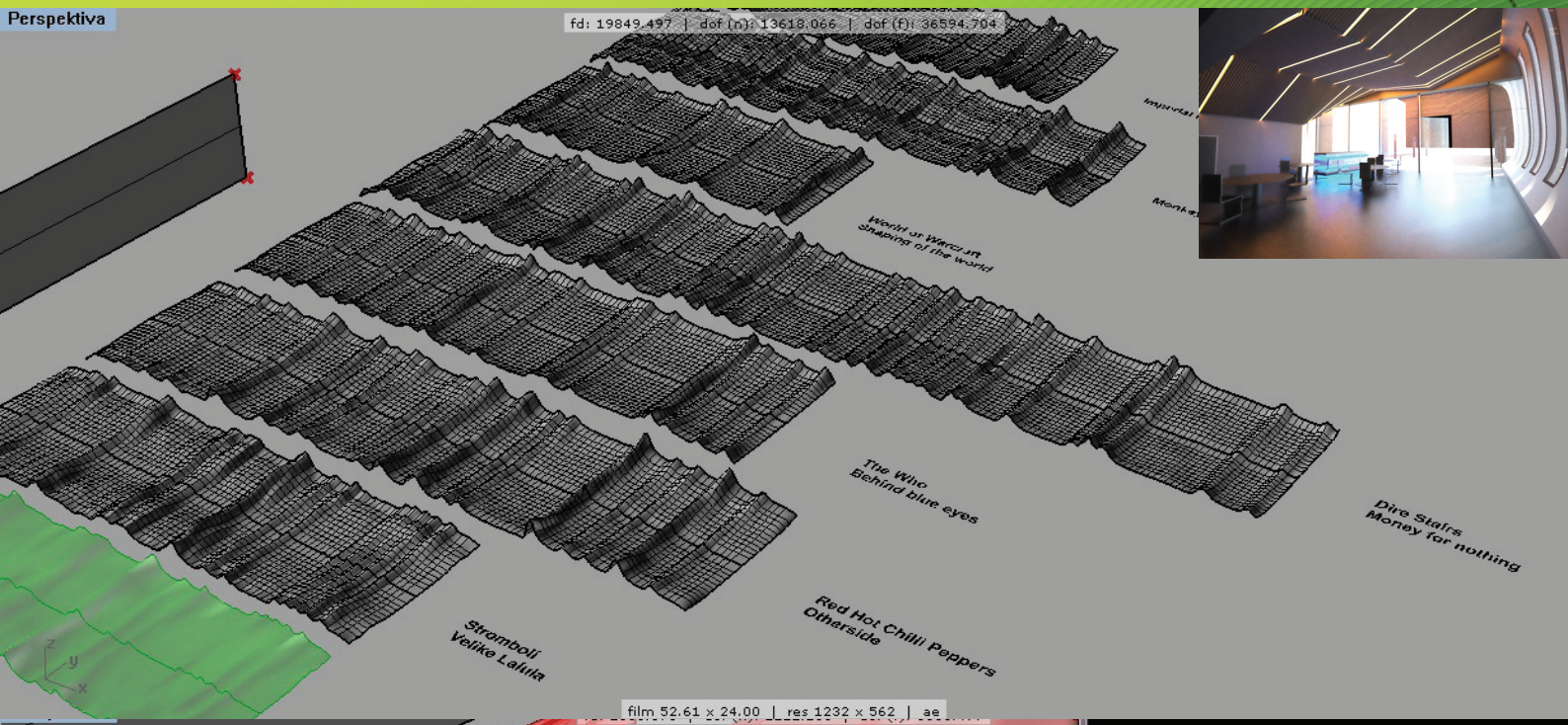
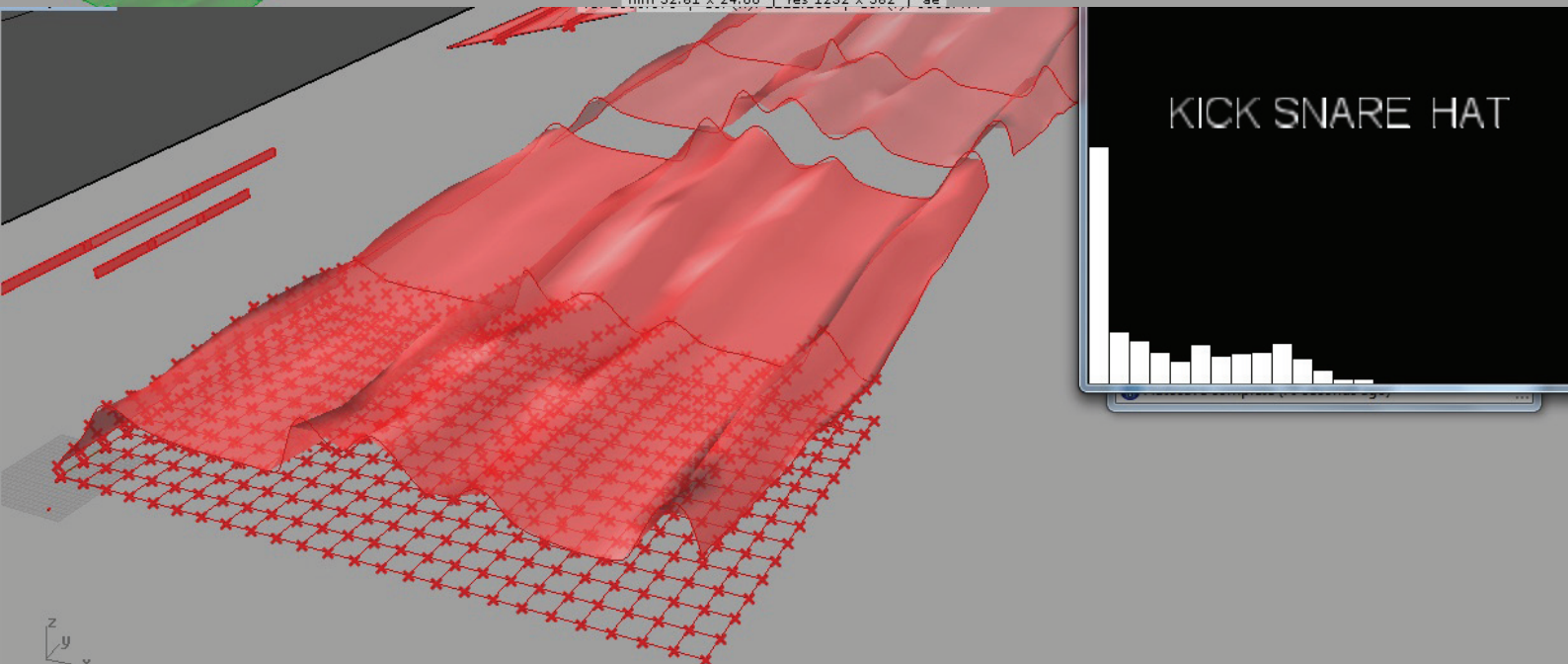
CAD IV - Scripting
FA - ČVUT 2009/2010
Jan Šmída

← Processing

Grasshopper





Analýza Audia - výstup dat Processing

http://processing.org/about/

Pro analýzu audia jsem využil program Processing , který má vv sobě mimo grafického rozhrani zaimplementovanou knihovnu minim Sound Libary, která má již připravené funkce pro práci se zvukem.
Pro analýzu jsem využil 2 připravené příklady :

- Frequency Energy
- Linear Averages

Jejich kombinací jsem vytvořil program, který generuje číselné řady s hodnotou ekvalizétu a násobiče které vycházejí z algoritmu Frequency energy.

Poznámka: Všechny 3 soubory je nutné po každém framu uložit, aby je mohl grasshopper načíst.

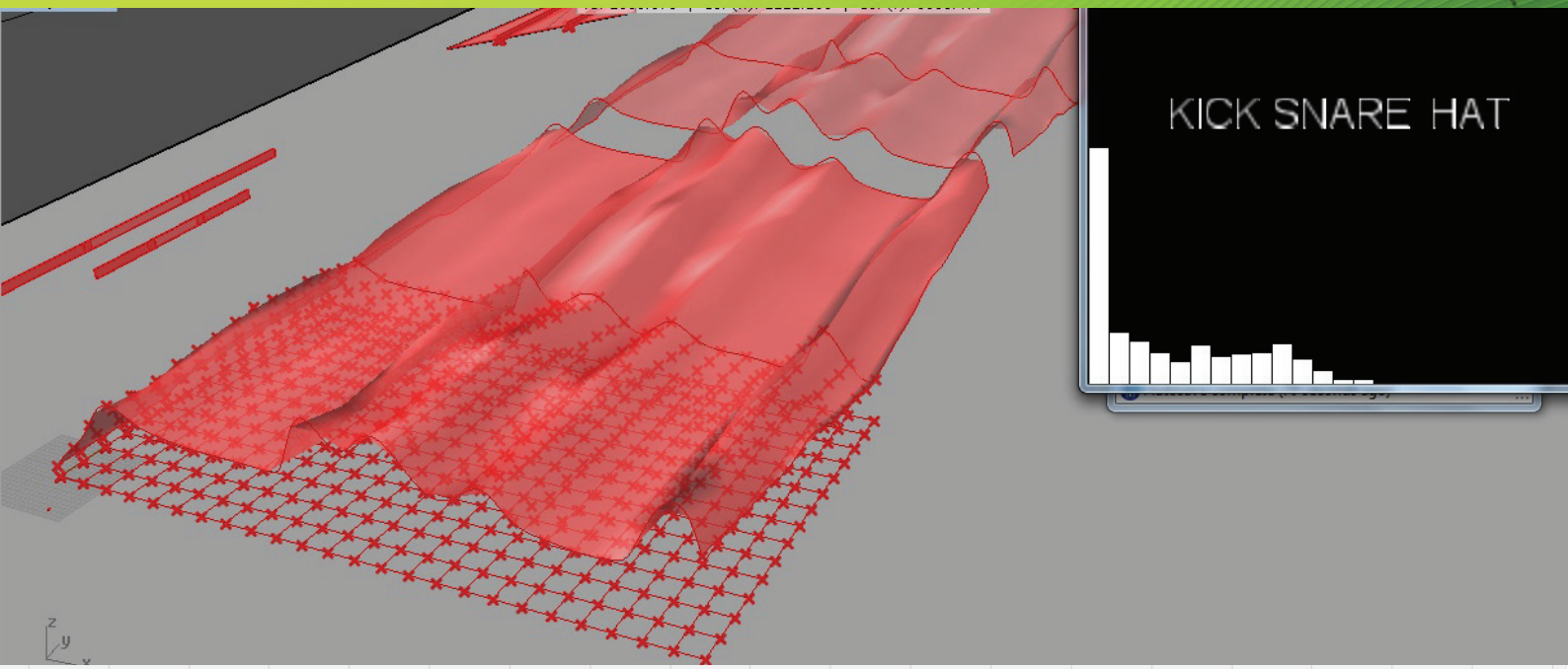
```
// Načtení externách knihoven
import ddf.minim.analysis.*;
import ddf.minim.*;
Minim minim;
AudioPlayer jingle;
FFT fft;
BeatDetect beat;
BeatListener bl;
// počet sloupců ekvalizeru
int pocet_sloupcu = 16;
// nastavení programu, proběhne pouze při spuštění.
void setup()
{
size(400,400,P2D);
minim = new Minim(this);
frameRate(25);
  // skladba
  jingle = minim.loadFile("monkey.mp3", 2048);
  // loop the file
  jingle.loop();
  // create an FFT object that has a time-domain buffer the same size as jingle's sample buffer
  // and a sample rate that is the same as jingle's
  // note that this needs to be a power of two
  // and that it means the size of the spectrum will be 1024.
  // see the online tutorial for more info.
  fft = new FFT(jingle.bufferSize(), jingle.sampleRate());
  // use 128 averages.
  // the maximum number of averages we could ask for is half the spectrum size.
  fft.linAverages(pocet_sloupcu);
  rectMode(CORNERS);

  beat = new BeatDetect(jingle.bufferSize(), jingle.sampleRate());
  beat.setSensitivity(300);
  kickSize = snareSize = hatSize = 16;

  bl = new BeatListener(beat, jingle);
  textFont(createFont("SanSerif", 16));
  textAlign(CENTER);
}

int rows = pocet_sloupcu;
int obnov = 50;
int vel_pole = 10000;
float[] prumer = new float[rows];
int okno = 1;
int pocet_radku = 0;
float nasobitel = 1;
float kickSize, snareSize, hatSize;
float[] soubor1 = new float[vel_pole];
float[] soubor3 = new float[vel_pole];
int soubor1_pozice = 0;
int soubor3_pozice = 0;
// funkce draw se provádí několikrát za sekundu podle fpsrate
void draw()
{
background(0);
fill(255);

  if ( beat.isKick() ) kickSize = 32;
  if ( beat.isSnare() ) snareSize = 32;
  if ( beat.isHat() ) hatSize = 32;
  textSize(kickSize);
  text("KICK", width/4, height/2);
  textSize(snareSize);
  text("SNARE", width/2, height/2);
  textSize(hatSize);
  text("HAT", 3*width/4, height/2);
  kickSize = constrain(kickSize * 0.95, 16, 32);
```

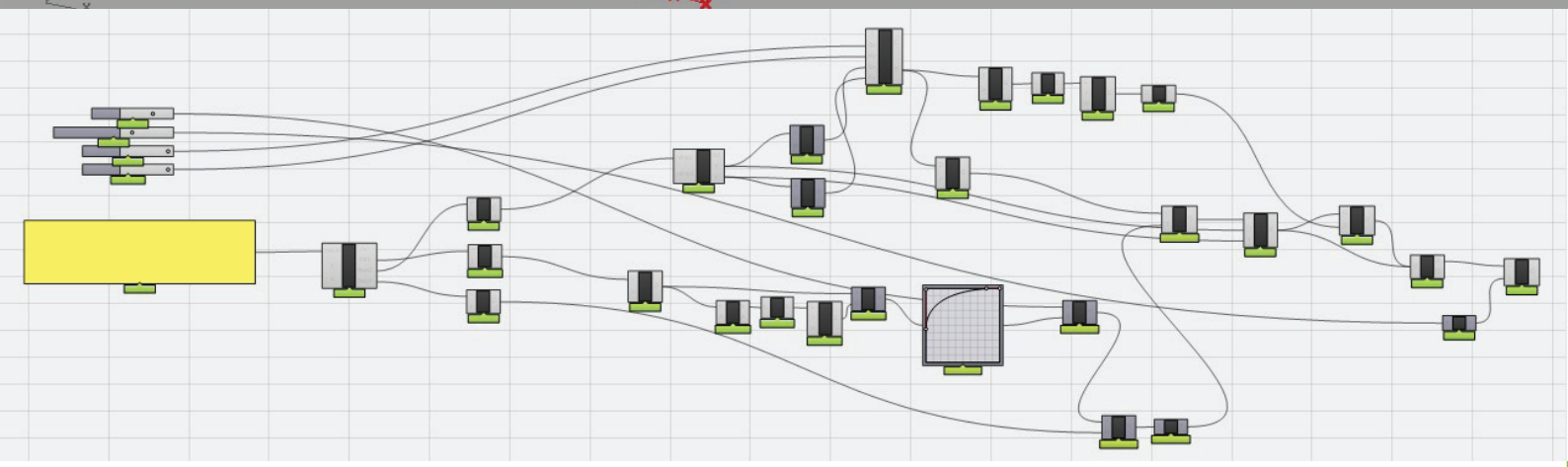
```
  snareSize = constrain(snareSize * 0.95, 16, 32);
  hatSize = constrain(hatSize * 0.95, 16, 32);

  fft.forward(jingle.mix);
  int w = int(fft.specSize()/64);
  for(int i = 0; i < fft.avgSize(); i++)
  {
    // draw a rectangle for each average, multiply the value by 5 so we
can see it better
    rect(i*w, height, i*w + w, height - fft.getAvg(i)*5);
    prumer[i] = prumer[i] + fft.getAvg(i);
  }
  okno++;
  if (okno > obnov)
  {
    okno = 1;
    for (int i = 1; i < pocet_sloupcu; i++)
    {
    soubor1_pozice++;
    soubor1[soubor1_pozice] = prumer[i]/obnov;

    if ( beat.isKick() ) nasobitel = nasobitel +1;
    if ( beat.isSnare() ) nasobitel  = nasobitel +1;
    if ( beat.isHat() ) nasobitel  = nasobitel +1;
    soubor3_pozice++;
    if(soubor3_pozice > 6)
    {
    nasobitel = soubor3[soubor3_pozice-1] + soubor3[soubor3_
pozice-2] + soubor3[soubor3_pozice-3] + soubor3[soubor3_pozice-4]
+ soubor3[soubor3_pozice-5] + nasobitel;
    }
    soubor3[soubor3_pozice] = nasobitel/6;
    nasobitel = 1;
    prumer[i] = 0;
    }
    pocet_radku++;

  PrintWriter output;
  PrintWriter output2;
  PrintWriter output3;
  output = createWriter("data.txt");
  output2 = createWriter("data2.txt");
  output3 = createWriter("data3.txt");
  for(int i = 1; i <= soubor1_pozice; i++)
  {
  output.println(soubor1[i]);
  }
 for(int i = 1; i <= soubor3_pozice; i++)
  {
  output3.println(soubor3[i]);
  }
  output2.println(pocet_radku);
  output2.println(pocet_sloupcu -1);
  output.close();
  output2.close();
  output3.close();
  }
}

void stop()
{
  // always close Minim audio classes when you finish with them
  jingle.close();
  minim.stop();
  super.stop();
}
```

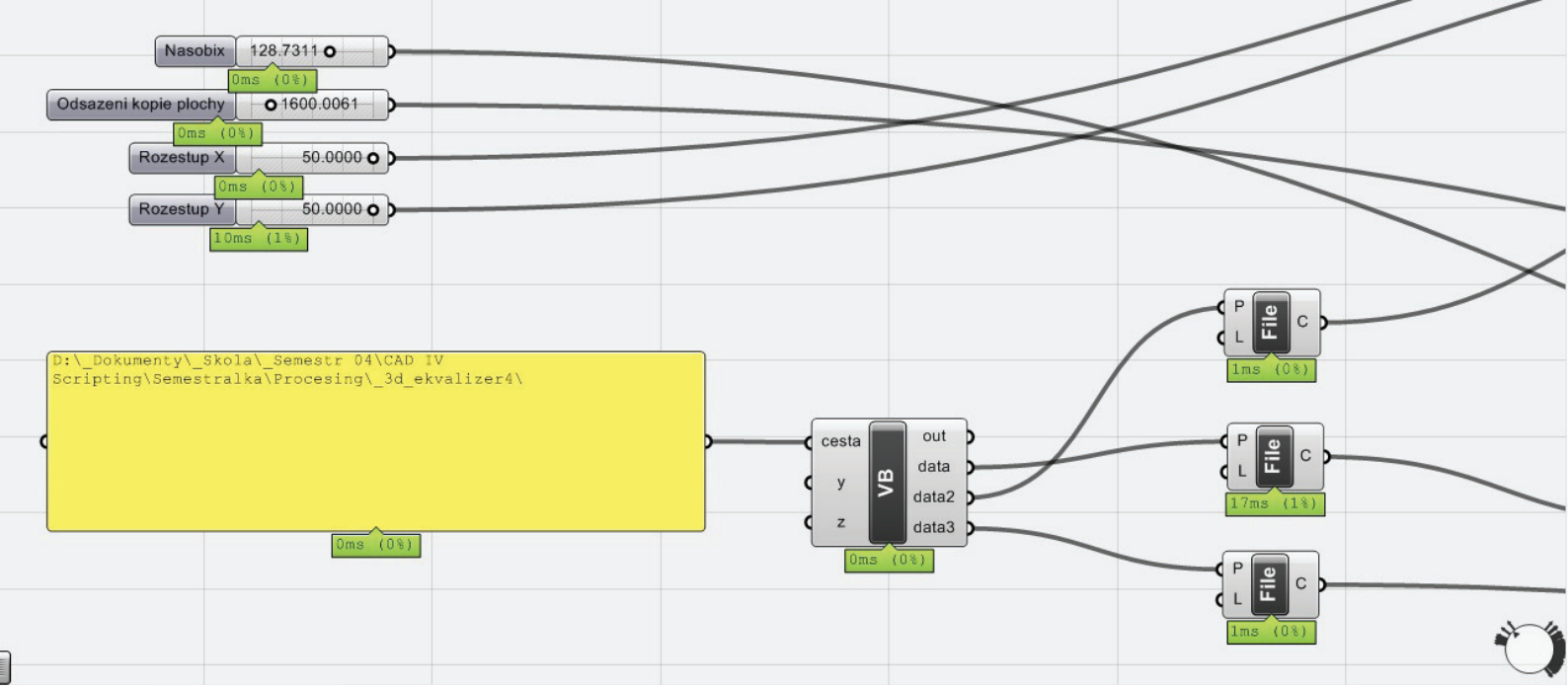
```
class BeatListener implements AudioListener
{
  private BeatDetect beat;
  private AudioPlayer source;

  BeatListener(BeatDetect beat, AudioPlayer source)
  {
    this.source = source;
    this.source.addListener(this);
    this.beat = beat;
  }

  void samples(float[] samps)
  {
    beat.detect(source.mix);
  }

  void samples(float[] sampsL, float[] sampsR)
  {
    beat.detect(source.mix);
  }
}
```

Generátor plochy Grasshopper

http://www.grasshopper3d.com/







Posuvníky:
- rozestupy x a y mřížky bodů
- násobič
- posun kopie plochy

cílový adresář

K cestě přidá názvy souborů pro komponenty:
<cesta>\<nazev souboru>

Komponenty načtou data ze souborů:
data.txt
data2.txt
data3.txt
řádek = list item

1.
Generátor 2D mřížky bodů, vygeneruje pravoúhlou mřížku bodů počet se určuje ze souboru data2.txt, rozměry mřížky se berou ze sliderů.
VB:

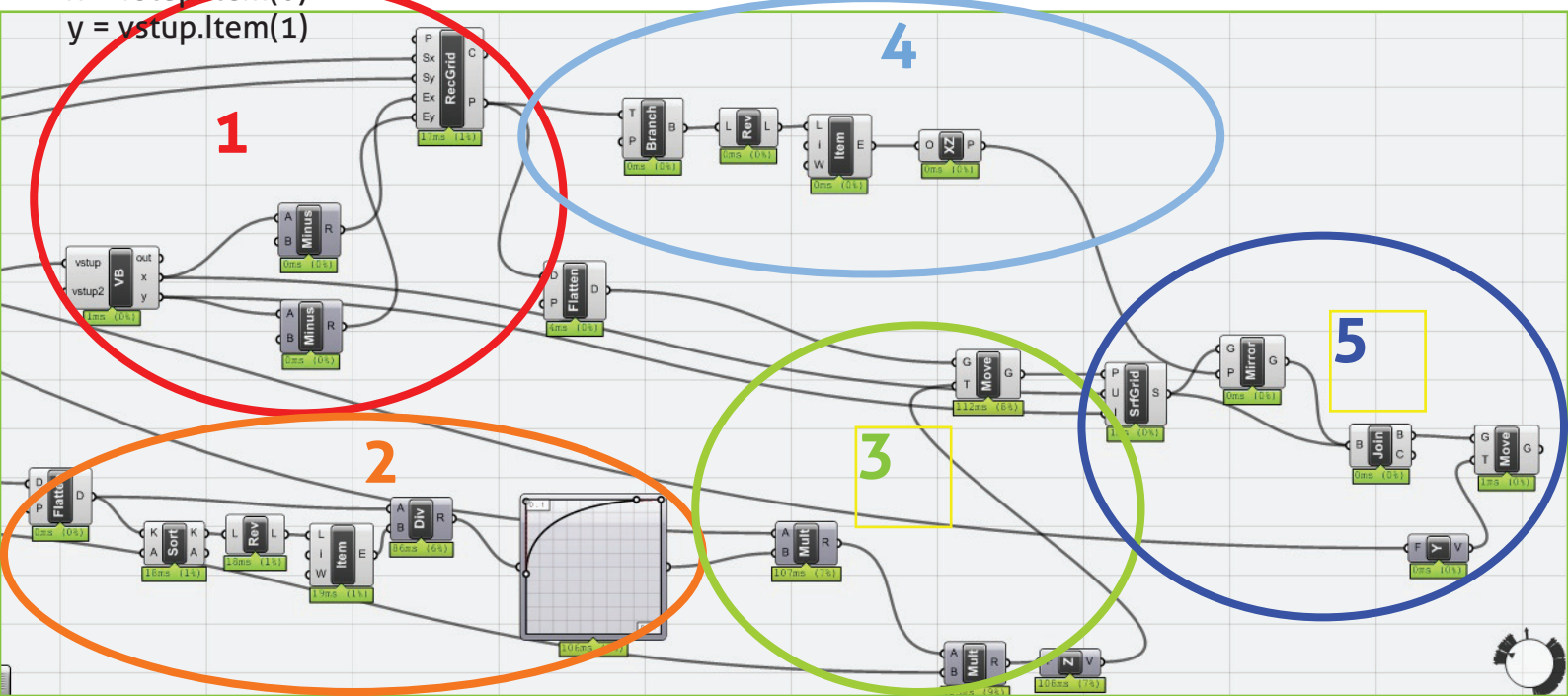
2.
Změní rozptyl hodnot ze souboru data.txt, tak aby čísla byla z intervalu <0,1>, dále je možné upravovat výstup hodnot pomocí křivky.

3.
Vynásobí data z algoritmu 2 hodnotami ze souboru data3.txt a honotami ze slideru nasobic. O výslednou hodnotu posune body z mřížky ve směru osy Z.

4.
Z množiny bodů plochy vybere krajní a do něho vloží rovinu, podle které se celá plocha odzrcadlí.

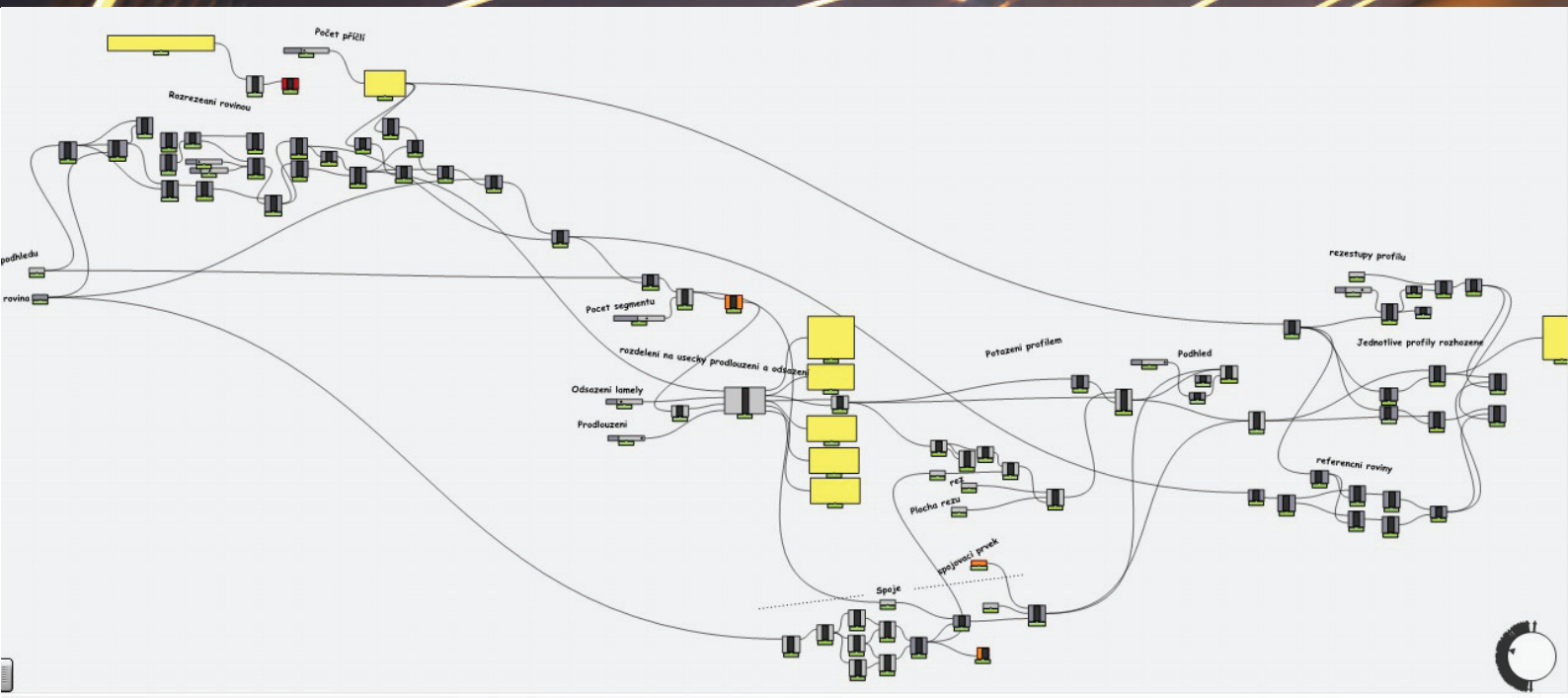
5.
Odzrcadlí plochu, spojí ji a posune výsledek o kus vedle.

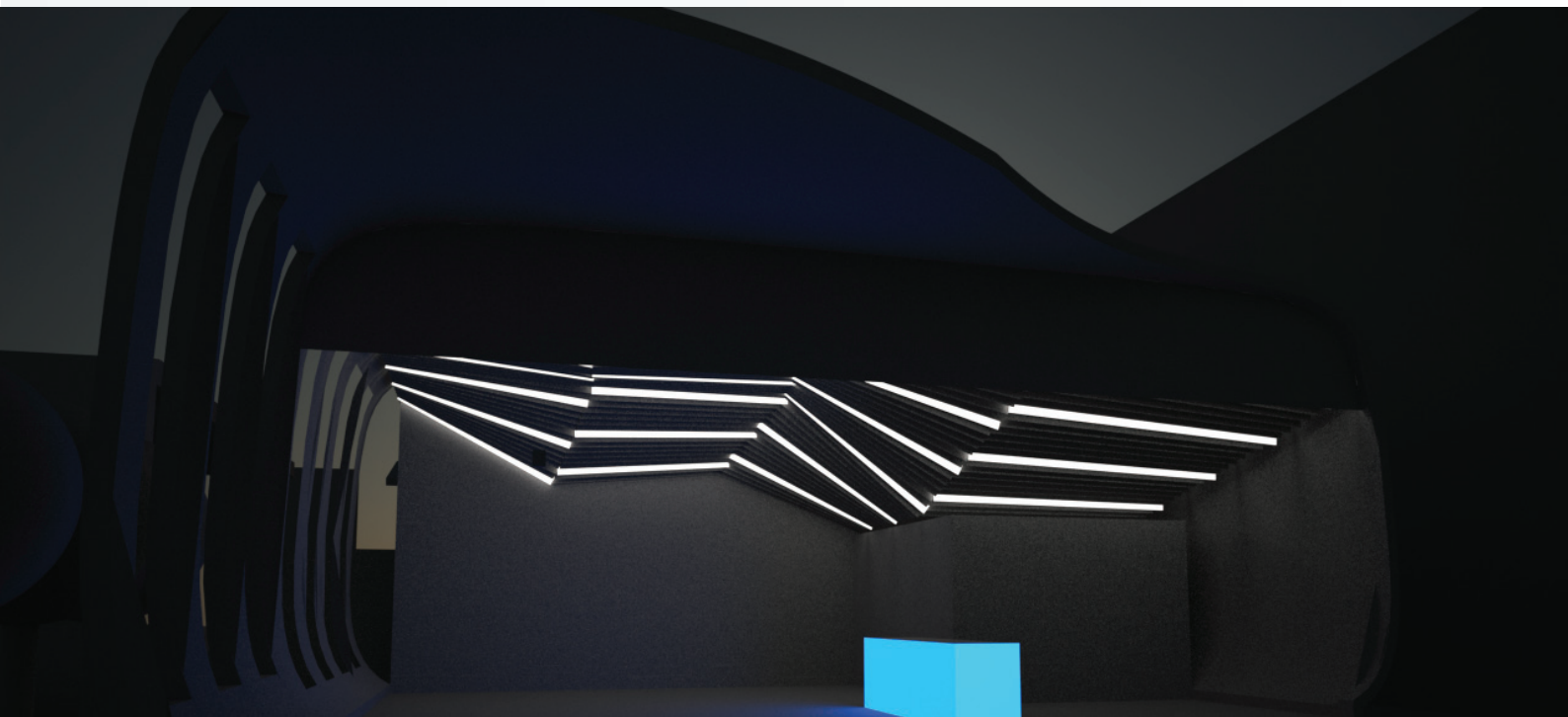
```
Private Sub RunScript(ByVal vstup As List(Of String), ByVal vstup2 As List(Of Double), ByRef x As Object, ByRef y As Object)
    'your code goes here...
    x = vstup.Item(0)
    y = vstup.Item(1)
```

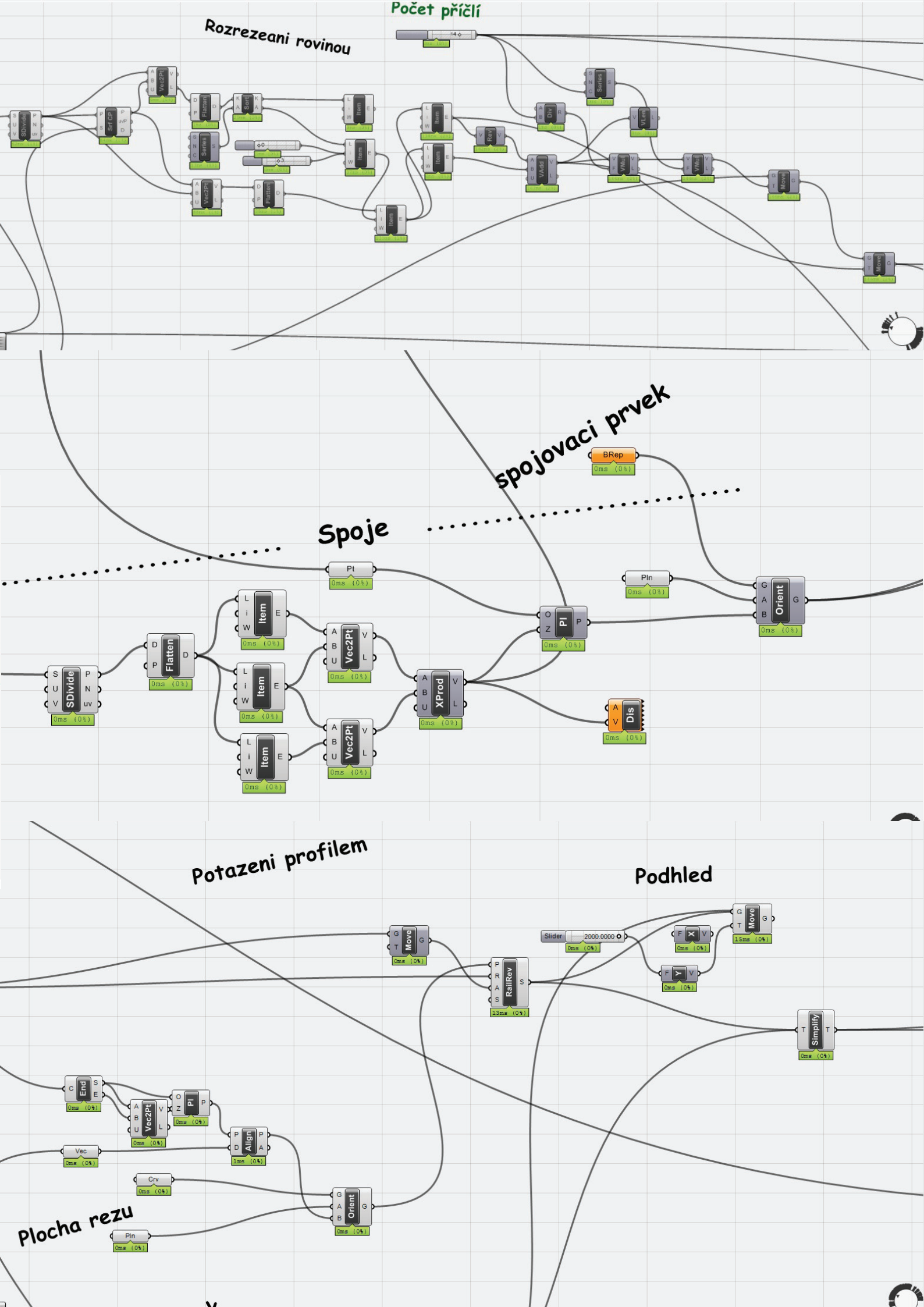


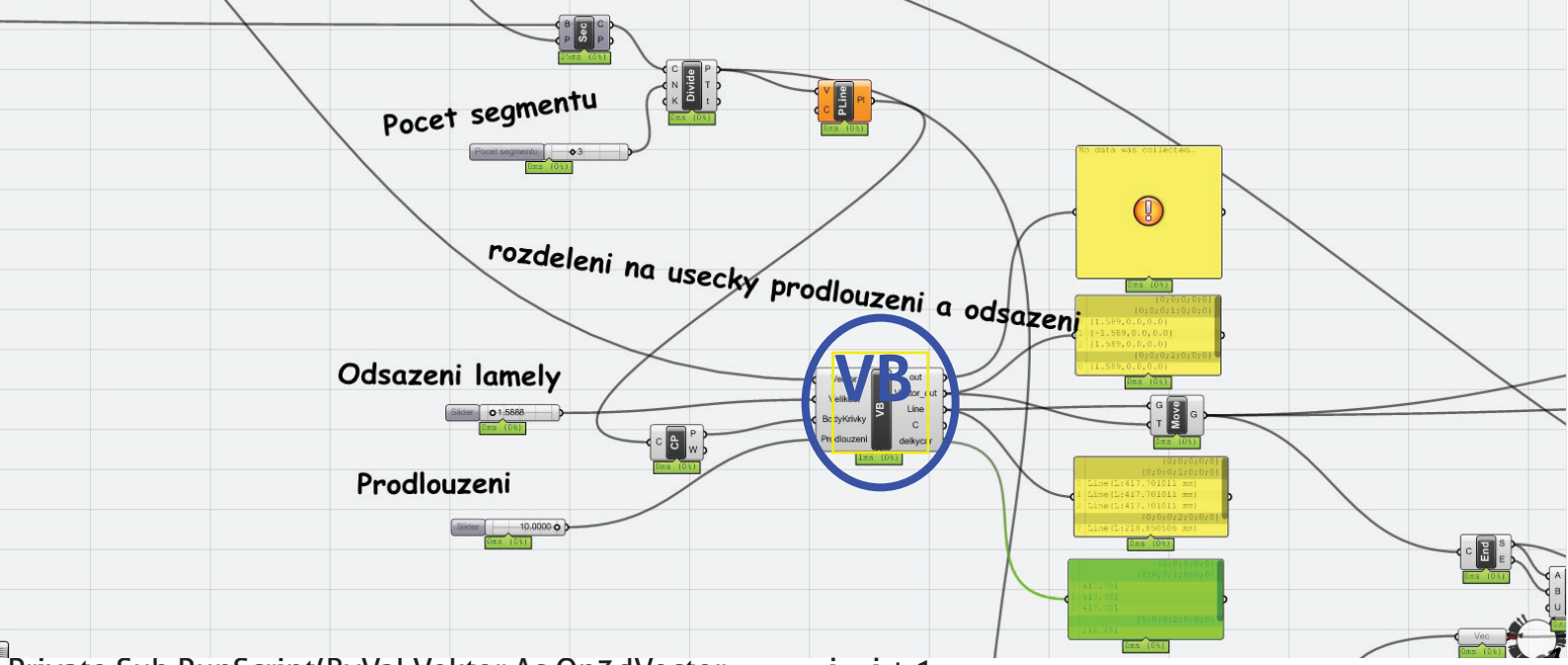
Parametrický podhled Grasshopper

http://www.grasshopper3d.com/






Počet příčlí
Rozrezeani rovinou
spojovaci prvek
Spoje
Potazeni profilem
Podhled
Plocha rezu
BRep
Pt
Pln
Pl
Orient
SDivide
Flatten
Item
Vec2Pt
XProd
Dis
Move
RailRev
Slider
Simplify
End
Vec
Align
Crv




```
Private Sub RunScript(ByVal Vektor As On3dVector, ByVal Velikost As Double, ByVal BodyKrivky As List(Of On3dPoint), ByVal Prodlouzeni As Object, ByRef Vektor_out As Object, ByRef Line As Object, ByRef C As Object, ByRef delkycar As Object)
  'your code goes here...
  'Dim Vektor_plosny As New On3dVector
  Dim Prepona As New Double
  Dim cary_pos As New List(Of OnLine)
  Dim delky As New List(Of Double)
  'Dim cara As New OnLine
  Dim cara_vektory As New List(Of On3dVector)
  Dim i As New Integer
  Dim j As New Integer
  Dim cisla As New List (Of Integer)


  j = -1
  i = 0

  While i < BodyKrivky.Count - 1
    Dim cara As New OnLine
    Dim Vektor_plosny As New On3dVector

    cara.from = BodyKrivky(i)
    cara.To = BodyKrivky(i + 1)
    cary_pos.Add(cara)
    Prepona = math.Sqrt(math.Pow(vektor.x, 2) + math.Pow(vektor.y, 2) + math.Pow(vektor.z, 2))
    Vektor_plosny.x = (j) * Vektor.x * Velikost / Prepona
    Vektor_plosny.y = (j) * Vektor.y * Velikost / Prepona
    Vektor_plosny.z = (j) * Vektor.z * Velikost / Prepona
    cara_vektory.add(Vektor_plosny)
    cisla.Add(i)

    prodluz(BodyKrivky(i), BodyKrivky(i + 1), Prodlouzeni, cara)
    C = cara
    delky.add(cara.Length)
    j = -j
    i = i + 1
  End While

  Vektor_out = cara_vektory
  Line = cary_pos
  delkycar = delky


End Sub

'<Custom additional code>


  Private Sub prodluz(ByVal A As On3dPoint, ByVal B As On3dPoint, ByVal delka As Double, ByVal cara As Online)
    Dim vektor As New On3dVector
    Dim pomer As New Double
    Dim AA As New On3dPoint
    Dim BB As New On3dPoint

    vektor.x = B.x - A.x
    vektor.y = B.y - A.y
    vektor.z = B.z - A.z
    pomer = delka / vektor.Length
    BB.x = B.x + pomer * vektor.x
    BB.y = B.y + pomer * vektor.y
    BB.z = B.z + pomer * vektor.z
    AA.x = A.x - pomer * vektor.x
    AA.y = A.y - pomer * vektor.y
    AA.z = A.z - pomer * vektor.z

    cara.from = AA
    cara.To = BB
  End Sub
```