STEP01 • GRID ON SURFACES

+ **Draw three lines** [Rhino input]
  - Note line direction

+ **Crv object [GH object]**
+ **Connect Rhino & GH objects**
  - RMB, select ‘set multiple curves’
  - Note line picking order

+ **Do the same thing for upper layer**

+ **Loft object**
  - Default loft option
+ Divide object for surface division

+ Number slider setting
  - ‘Even numbers’ for U direction
  - ‘Integer’ for V direction

+ Connect Divide object to surfaces & sliders
By default, points generated through ‘Divide Srf’ object are to be ordered in zigzag fashion.

For better control, point ordering method should be changed into 2 x 2 array or list.

Get mid points
- pt01A = (ptA(0)(0) + ptA(1)(0)) / 2
- pt02A = (ptA(0)(1) + ptB(0)(1)) / 2
- pt03A = (ptA(2)(0) + ptA(1)(2)) / 2
- pt01B = (ptB(0)(0) + ptB(1)(0)) / 2
- pt02B = (ptA(1)(0) + ptB(1)(0)) / 2
- pt03B = (ptB(2)(0) + ptB(1)(2)) / 2

Set direction vectors
+ Draw two interpolate curves.

+ Loft

+ Component expansion
+ Point List object attached to see point order on surfaces (zigzag order)

+ Setting VB component
  - Four input parameters
  - ptSetA: points set A from upper surface
  - ptSetB: points set B from lower surface
  - divU: U direction division factor
  - divV: V direction division factor

+ Get connected

+ Set input parameters: ptSetA & B
  - Check 'List' - multiple point input
Set input parameters: ptSetA & B
- Check ‘On3dPoint’ for data type hint

Set input parameters: divU & V
- Check ‘Integer’

Double ‘For ~ Next’
- To remap one dimensional linear point input data into two dimensional array or list, we will use double ‘for ~ next’ loop.

Iteration for V direction (‘j’ direction, pt01 ~ pt06)
- Define ‘n’ as integer. (Overall index, 0 to point upper bound)
- Define ‘ptRowA’ as list of points
- Define ‘ptA’ as individual points
- Assign the point ‘ptSetA(n)’, n th member of input point list, onto ‘ptA’, temporary address
- Add the point on ‘ptRow’
- Increase ‘n’ by 1
• Iteration for U direction (‘i’ direction, pt07 ~ pt13 / pt14 ~ pt20 / ... / pt28 ~ pt34)
  - Define ‘ptListA’ as list of list (not list of points)
  - Add ‘ptRow’ to ‘ptList’

• Duplicate for the lower surface points
  - Duplicate codes for ‘ptSetB’

• Double ‘For ~ Next’ for point assigning
  - Since our component is 2(U) by 1(V), set U direction step as 2

• Get mid points
  - Note that 4th point is to get direction vectors
Get U directional module distances and directional vectors to unitize starting and ending vectors

Since we need to draw 'interpolate curves', look up Rhino .NET SDK help file.
- Rhino4DotNetPlugIns

To draw interpolate curves, we should define point array first.
- In this case, the point array should contain three points defined in the previous step except point #4. Remember the 4th point was just to get vectors.
- Note that On3dPointArray is not same with Array of On3dPoint.

Define new list of Nurbs curve
Define On3dPointArray.
Append three points to the Array
Define interpolate curve as a nurbs curve
Draw interpolate curve with argument (3, ptArrA, vt01A, vt02A, Nothing)
Add the curve to interpolate curve list

Set output as the list of interpolate curves
Duplicate each codes for the lower surface
Loft using two sets of interpolate curves
- Unexpected loft result

Curves on the same data branch

Grafting
- Creates a new branch for every single data item.

Lofting
+ Modify input curves

+ Or control div numbers