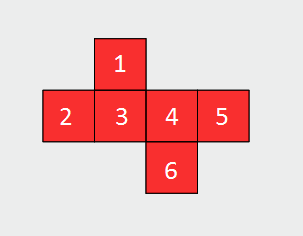
# 3D NETS

* Entire code available at **folding-nets-2015-01-02-a** GIT branch.  
  Interactive name - shape-folding
* Folding animation implemented using Cango-3D library. (<http://www.arc.id.au/Canvas3DGraphics.html>)
* Drag drop of different shapes implemented using Cango-3D library.
* Snapping of shapes currently done using Paper JS (<http://goo.gl/kopJlL>). To be converted to Cango-3D.  
  Code present in ***shape-folding/js/views/snapping.js***.
* Structure implemented for folding animation - Depth First Tree Traversal  
  

Folding carried out as follows:  
Root node: **3**  
Folding Sequence: **[4,5,6]**, **[5]**, **[6]**, **[1]**, **[2]**

* Currently a ***nets-data.json*** file is used to stored predefined nets. Later to be generated using drag-drop and snapping of shapes.
* Current data allows only square faces. To be extended for triangular and rectangular faces.
* Format of data stored -

{

"faces": [

[ 0, 0 ],

[ 1, 0 ],

[ 1, 1 ],

[ 2, 1 ],

[ 3, 1 ]

],

"solution": [

{

"faces": [ 4, 5 ],

"side": 3,

"angle": 90

},

{

"faces": [ 5 ],

"side": 3,

"angle": 90

},

{

"faces": [ 2, 1 ],

"side": 2,

"angle": 90

},

{

"faces": [ 1 ],

"side": 1,

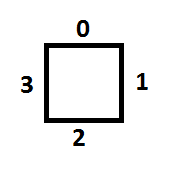
"angle": 90

}

]

}

* faces -  
  Top and left position point factors of each square face (to be extended for rectangular and triangular faces)
* solution.faces -  
  List of faces to be rotated (later to be added in a group).
* solution.side -  
  Side of the first face from the above array about which the above faces are to be rotated. Below is the numbering used.



* solution.angle -  
  Angle to be rotated about the above side along the z-axis.