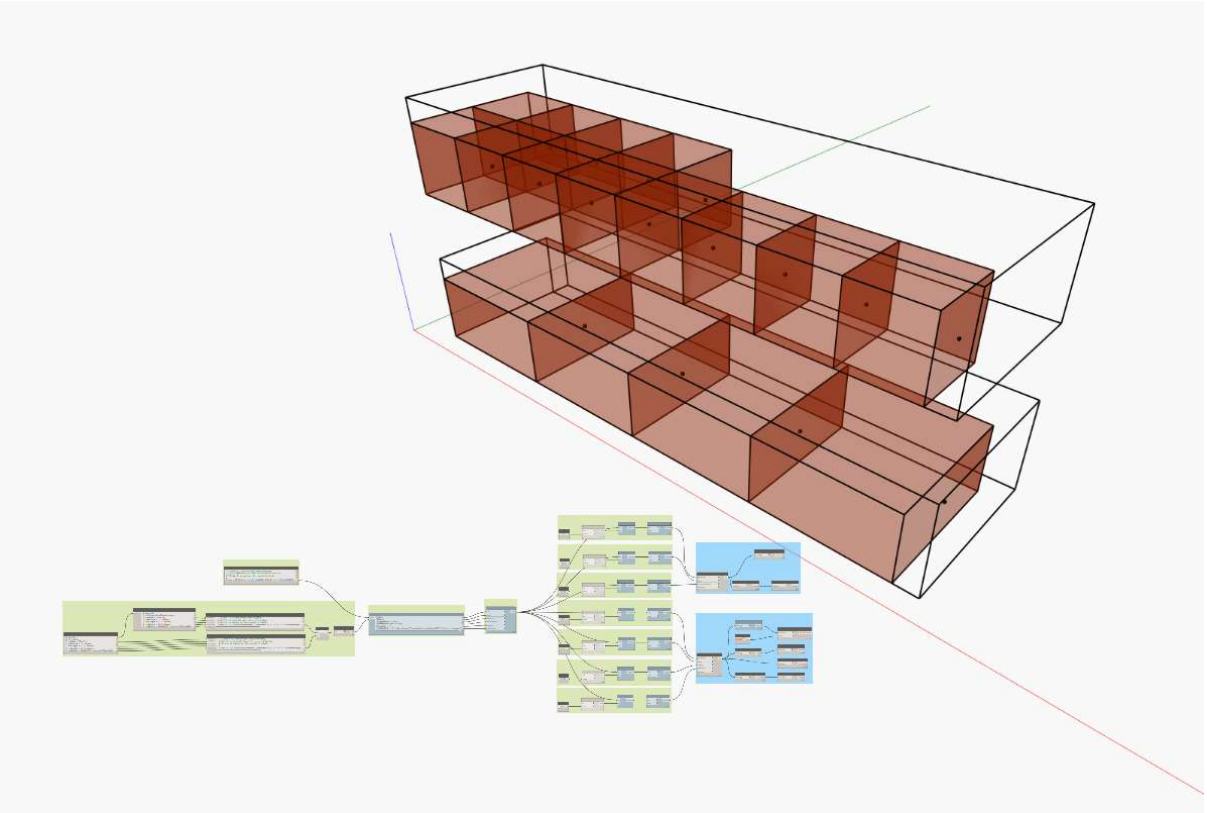
3D Bin Packing

Instructions



Introduction

3D Bin Packing allows you to define one or more axes-aligned *bins* (cuboids) in which to pack one or more *items* (cuboids).

Prerequisites

You should have the latest version of Dynamo that can use CPython3

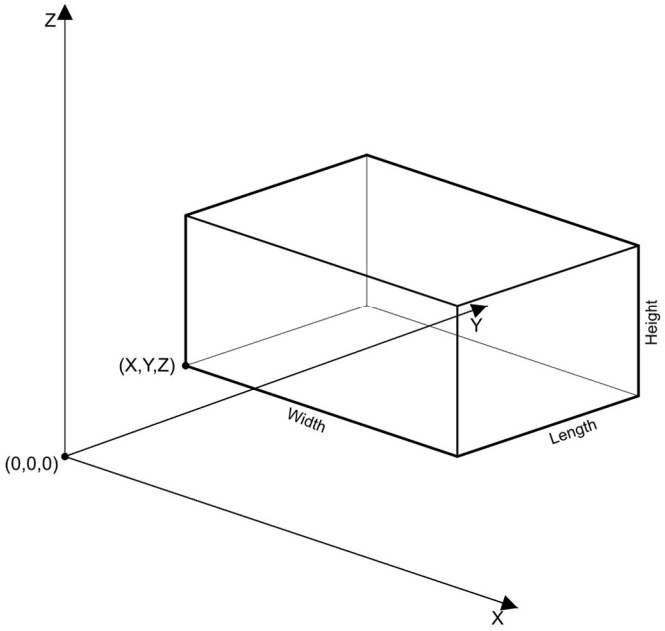
You should install p3dbp from https://pypi.org/project/py3dbp/

You should install Topologic either from the package manager or from https://topologic.app

Make note of its installation folder to input into the definition

We use a left-handed coordinate system and Z-Axis is assumed up.

Create a Bin



A bin is assumed to be an axis-aligned cuboid and defined as a list: [*binName*, *X*, *Y*, *Z*, *Width*, *Length*, *Height*, *MaximumWeight*] e.g. [“bin1”,0,0,0,10,20,30,1000]. If you wish to use more than one bin then you create a list of bins (e.g. [[“bin1”,0,0,0,10,20,30,1000], [“bin2”,0,0,30,5,10,15,500]]). Bins can be located anywhere, have different sizes, and different weight capacities.

- *binName* (string) is an arbitrary name of the bin. It does not have to be unique, but it is advised to be unique for easier identification at the end of the process.
- *X* (double) is the location of the bin’s lower-left corner on the X-axis
- *Y* (double) is the location of the bin’s lower-left corner on the Y-axis
- *Z* (double) is the location of the bin’s lower-left corner on the Z-axis
- *Width* (double) is the width of the bin (parallel to the X-axis)
- *Length* (double) is the length of the bin (parallel to the Y-axis)
- *Height* (double) is the height of the bin parallel to the Z-axis
- *MaximumWeight* (double) is the maximum total weight of packed items that the bin can accommodate. A bin will stop packing items if it reaches this maximum weight even if it has more space in it.

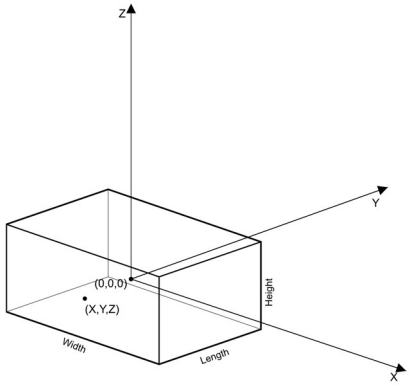
Create Bins

Code Block

```
//[name,X,Y,Z,width,length,height,maxWeight]
//X,Y,Z represent the lower left corner of the bin;
//Z-axis is assumed up. Corresponds to height;

bins = [["binA",2,1,0,40,7,7,1000], ["binB",2,1,7,40,7,7,1000]];
```

Create an Item



An item is assumed to be an axis-aligned cuboid centred around the origin (0,0,0). An item is assumed to have an entry point associated with it located at an arbitrary location. The entry point does not affect the packing in any way but can give a visual indication regarding the rotation of the packed item in the bin. An item is defined as a list: [*itemName*, *X*, *Y*, *Z*, *Width*, *Length*, *Height*, *Weight*]. If you wish to pack more than one item then you can create a list of items (e.g. [[“item1”,-5,0,0,5,5,6,100], [“item2”,10,5,2,2,2,3,55]]). Items can have different entry points, different sizes, and different weights.

- *itemName* (string) is an arbitrary name of the bin. It does not have to be unique, but it is advised to be unique for easier identification at the end of the process.
- *X* (double) is the location of the bin’s lower-left corner on the X-axis
- *Y* (double) is the location of the bin’s lower-left corner on the Y-axis
- *Z* (double) is the location of the bin’s lower-left corner on the Z-axis
- *Width* (double) is the width of the bin (parallel to the X-axis)
- *Length* (double) is the length of the bin (parallel to the Y-axis)
- *Height* (double) is the height of the bin parallel to the Z-axis
- *MaximumWeight* (double) is the maximum total weight of packed items that the bin can accommodate.

Pack Items in Bins

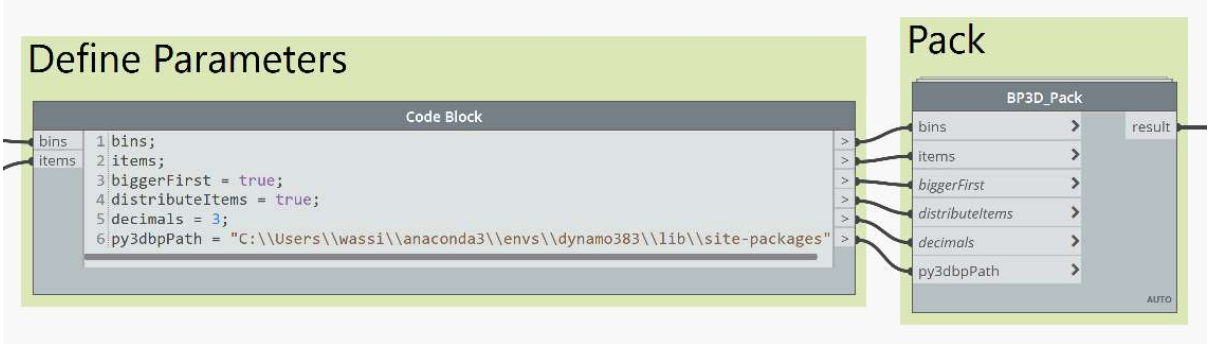
Prepare a list of bins and items as described above. Drag and drop the node titled ***BP3D_Pack***. Enter the following information:

- *bins* (list) list of bins
- *items* (list) list of items
- *biggerFirst* (bool – optional – default: true): If set to True then pack the bigger items first. Otherwise, pack the smallest items first.
- *distributeItems* (bool – optional – default: true): If set to True distribute the items across all bins. Otherwise, pack the same items into several bins (note: this duplicates items and can have unexpected and incorrect results if the bins are of different sizes).

- *decimals* (int – optional – default: 3): The number of decimal points to use. This improves accuracy.
- *py3dbpPath* (string): This is the folder location of the py3dbp python library.

The output of this node is a nested list structured as follows:

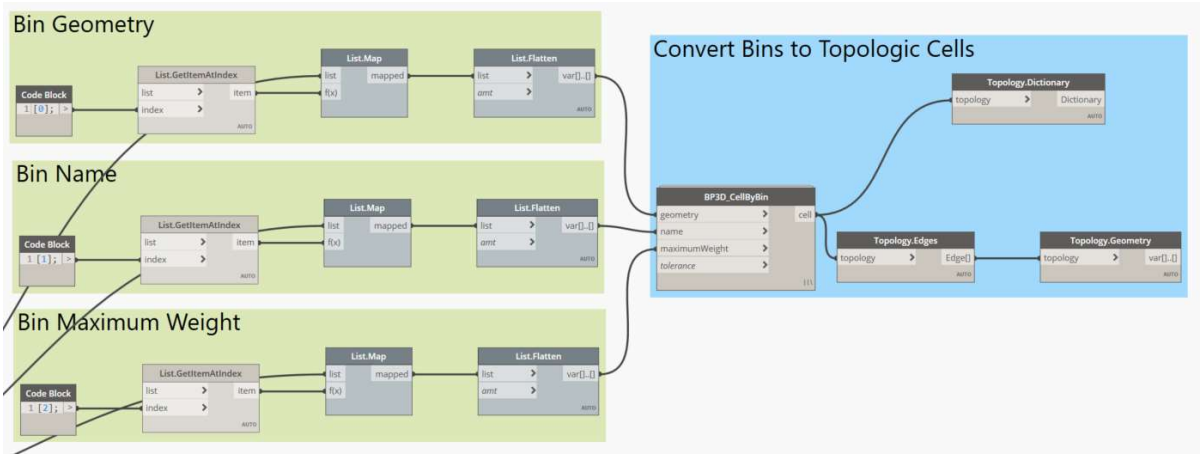
- *binGeometry* (geometry)
- *binName* (string)
- *binMaximumWeight* (double)
- *itemGeometries* (list)
- *itemNames* (list)
- *itemWeights* (list)
- *itemEntryPoints* (list)



Convert Bins into Topologic Cells

Drag and drop a ***BP3D_CellByBin*** node. This node converts a bin into a Topologic Cell and sets the correct Dictionary keys and values. This node takes as input the following:

- *geometry* (geometry – binGeometry)
- *name* (string – binName)
- *maximumWeight* (double – binMaximumWeight)
- *tolerance* (double – optional – default: 0.0001)



Convert Items into Topologic Cells

Drag and drop a ***BP3D_CellByPackedItem*** node. This node converts a packed item into a Topologic Cells and sets the correct Dictionary keys and values. The entry point is stored as a Topologic Vertex in the contents of the Topologic Cell. This node takes as input the following:

- *geometry* (geometry – itemGeometry)
- *name* (string – itemName)
- *weight* (double – itemWeight)
- *entryPoint* (Point – entryPoint)
- *tolerance* (double – optional – default: 0.0001)