Lego Assembly

Author: Romain Testuz
Supervised by: Yuliy Schwartzburg
Previous work

• All the approaches are based on a variation of the following heuristic:

\[
\text{Fitness} = C_{\text{numbricks}} \times \text{numbricks} + C_{\text{perpend}} \times \text{perpend} \\
+ C_{\text{edge}} \times \text{edge} + C_{\text{uncovered}} \times \text{uncovered} \\
+ C_{\text{otherbricks}} \times \text{otherbricks} + C_{\text{neighbour}} \times \text{neighbour},
\]

• They all try to prevent problems but they never actually try to find the problems to solve them.
Our approach

• Our method will not try to prevent problems, it will correct them.
• At the end we know if there are problems and we can estimate their seriousness.
Load in Dolphin
Pre-hollow

Voxelize

External program

Pipeline

Optimize
Finalize
Merge
Merge

Cost function: favors the merges that will create the most connections
Cost function: favors the merges that will create the most connections
Connectivity graph

Vertex <=> brick

Edge <=> connection
How to measure solidity?

1. All the bricks should be connected: connected graph
2. We do not want any brick to be the only connection between 2 large parts of the construction: articulation points
Connected components
Bad articulation points

- **Articulation point**: a vertex that if removed increases the number of connected components
- **Bad articulation point**: an articulation point which connects 2 or more subgraphs of size greater than 1
“Bad” articulation points
Removing connected components and bad articulation points
Using colors
Pre-hollow

• Reduces the number of bricks before optimization
• Very fast

Shell size = 2
Post-hollow

• Start when there is a single connected component and no more bad articulation point.

• Remove an inside brick only if it won’t introduce more connected components or bad articulation point.
1. Select a random brick in the inside of the model.
2. Create a 3-ring subgraph of the connectivity graph centered on this brick.
3. Compute the connected components and the articulation points of this subgraph.
4. Remove the vertex corresponding to the current brick.
5. Recompute the connected components and the articulation points.
6. If there are more connected components or more articulation points in the second subgraph, the brick cannot be removed. Otherwise, the brick can be removed and the connectivity graph updated.
Satisfying brick type limit

• Last step
• Similar to post-hollowing
• Select an extra brick and see if there is a cut that will not create weak points

The 4 cuts of a 2x6 brick
Instructions
Results
Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Layers</th>
<th>Voxels</th>
<th>w/o post-hollowing</th>
<th>With post-hollowing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lego man top</td>
<td>30</td>
<td>4’295</td>
<td>0.5</td>
<td>921</td>
</tr>
<tr>
<td>Cube</td>
<td>32</td>
<td>14’912</td>
<td>1.2</td>
<td>2’421</td>
</tr>
<tr>
<td>Stanford Bunny</td>
<td>53</td>
<td>21’393</td>
<td>3.7</td>
<td>5’350</td>
</tr>
<tr>
<td>Mario</td>
<td>64</td>
<td>16’735</td>
<td>3.4</td>
<td>4’222</td>
</tr>
<tr>
<td>Armadillo</td>
<td>110</td>
<td>138’393</td>
<td>25.3</td>
<td>16’201</td>
</tr>
</tbody>
</table>

- All of the above models have a single connected component and no bad articulation point.
Demo
Legal bricks

Can be changed except for the 1x1 and 1x2
Merging

1. Choose a random brick
2. Find the set of neighbours with which the brick can be merged
3. Select one of them (randomly or with a cost function) and merge
4. Goto 2. until there is no more mergeable neighbours
5. Goto 1