Procedural Modelling of Buildings

Pranjal Khare: 130050028
Charmi Dedhia: 130070007
Maitreyee Mulawkar: 13D170011
Anand Bhoraskar: 130050025

The problem statement

The task is to create a shape grammar, with production rules that iteratively creates coarse to fine representations of a building specified by the grammar. The grammar should also have a probabilistic aspect to be able to model variations of buildings.
This can be useful to create the environments for computer games and movies.

Implementation

The primary components of the implementation are:

- The parser
- The modeler
- The shape tree
- Rendering aspects

The parser required processing the grammar file and populating a rule set which was put inside the modeler. Using the grammar, we created a tree in which all the terminal symbols of the grammar which are at the leaf node comprise of renderable objects like wall, window, roof and door. This gave us a list of primitives that we needed to render.

We provide the following functions in our grammar:

Comp: Takes a 3d bounding box and divides it into it's corresponding lower dimensional components. Example: a building would have its walls as components

Subdiv: Takes a dimension and parameters and slices the shape along this dimension according to the parameters. Parameters provided can be **absolute** or **relative**

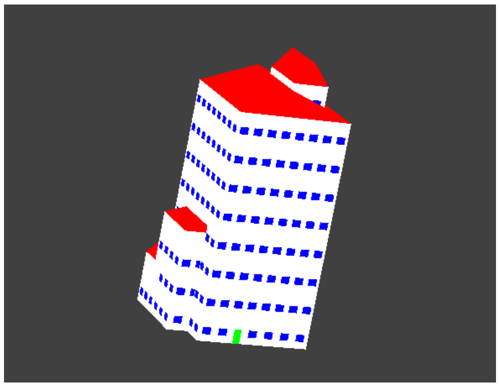
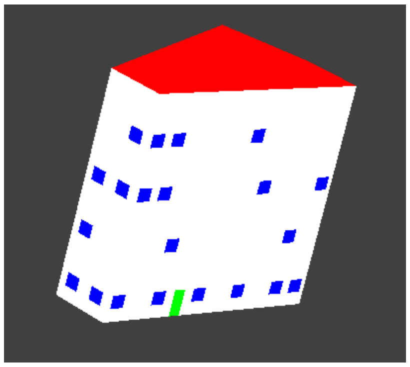
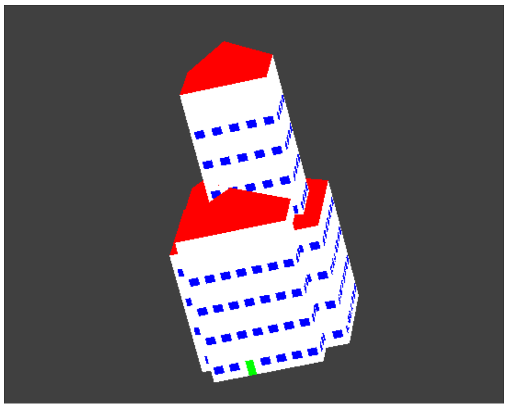
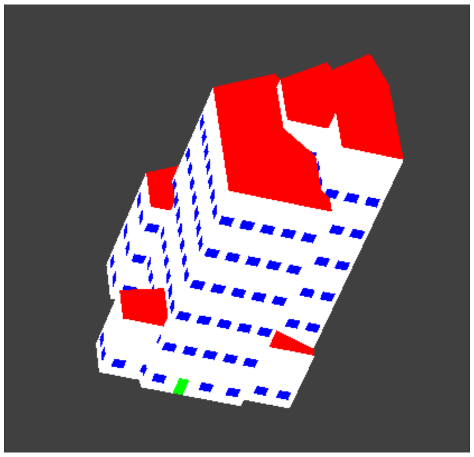
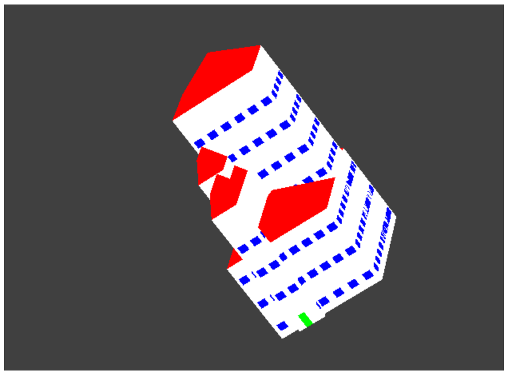
Repeat: Takes a dimension and repeats the child shape along this dimension enough number of times so as to fill the parent shape. Parameter can be **absolute** (left out space treated as a wall) or relative (left out space is distributed among the children)

MSB: Creates a convoluted building inside the bounding box by combining multiple randomly generated buildings according to number of children provided

S/T: For scaling and translation

Probabilities: We also provide a feature to assign probabilities to various rules. The rule would be invoked with the given probability

Results

Who did what?

This question has no straightforward answer. However all of us usually worked for roughly the same time. And each component was worked upon by each one of us at some point of time.