Procedural House Generation
A method for dynamically generating floor plans

Jess Martin
http://www.cs.unc.edu/~eve/research/procedural/

preliminaries
Christopher Alexander’s A Pattern Language
- private versus public room distinction
  Parameterization: Current parameters are
  - lot size and land value. They impact:
    - footprint size: exterior boundaries of the house
    - room size: average size of rooms in the house
    - public to private ratio: how much space on public

graph generation
The goal of this step is to create a graph of the floor plan where:
rooms = nodes (● Bedroom)
doors = edges (● — ●)
The graph is “grown” using a context-free grammar with the following rule-sets:
and a set of statistics for each room type:

room placement
The goal of this step is to place the nodes of the graph in 2D space.
This is accomplished using standard 2D graph layout methods for planar graphs. The method used here subdivides the available radial space between connected nodes, creating a spider-web appearance.

room expansion
The goal of this step is to expand each room to the appropriate relative size depending on the type of room it is.
This is done by assigning each room a “pressure” in relation to its relative size: larger rooms (like living rooms) exert more pressure than smaller rooms (like bathrooms). Then a room is randomly chosen to check the surrounding rooms and evaluate their pressures. It is then expanded or contracted accordingly.

results
The method is extremely fast and would be suitable for real-time applications:

Houses are (mostly) plausible, but would break down under close scrutiny. Especially by architects.

Useful algorithm for populating large simulations where buildings must be unique and contain quasi-realistic interiors.

Improvements:
- Roofs
- Multi-story
- Arch. styles
- Better expansion
- “Finishing”

Eventual goal is to create a highly parameterized design exploration tool.

*Actual generated model. Added textures, roof, windows, and door.*